PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS Y COMPUTACIÓN

DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE

INGENIERO DE SISTEMAS Y COMPUTACIÓN

TEMA

"GUÍA METODOLÓGICA PARA EL PROCESO ENSEÑANZA- APRENDIZAJE DE BIG DATA."

AUTORES

ANDREA CAROLINA RAMOS RAMÓN

JONATHAN FABRICIO LÓPEZ DÁVILA

QUITO, 2016

AGRADECIMIENTO

Ha sido un largo camino por recorrer, pero ahora al finalizar mi etapa como universitaria puedo decir que esta gran experiencia llena de altos y bajos, de buenos y malos momentos, de desafíos, aventuras y grandes aprendizajes, fue posible gracias al apoyo de personas muy valiosas que han estado a mi lado a lo largo de este tiempo.

Quiero agradecer principalmente a Dios por darme salud y vida, así como también agradecerle por brindarme la fortaleza y sabiduría necesarias para poder enfrentar tanto los retos como problemas que pudieron haberse presentado en el transcurso de este periodo universitario.

Agradezco a mis padres Mery y José por ser mis guías en cada paso que he dado, por sacrificarse a diario por brindarme la mejor educación y los recursos necesarios para que yo haya podido culminar con mis estudios, por sus grandes y sabios consejos que han sido mi pilar a lo largo de mi vida y sobretodo, por ser los mejores padres que Dios pudo haberme dado el gusto de tener. Al mismo tiempo, agradezco a mi familia por ser un apoyo emocional en cada momento.

También le doy las gracias a una persona muy especial para mí, a Jonathan, por haber compartido conmigo experiencias, emociones, alegrías, desafíos, dificultades y momentos importantes e inolvidables. Gracias por tus consejos, paciencia, apoyo y entereza, ya que con estas virtudes juntos logramos realizar y finalizar productivamente este trabajo de disertación. Además, gracias por haberme enseñando muchas cosas y por estar junto a mí en este largo camino.

Finalmente, quiero agradecer a mis mejores amigas Gabriela y Andrea por su amistad incondicional, apoyo, cariño y por estar junto a mí en los buenos y malos momentos. Y a mis maestros Suyana, Alfredo y Damián, ya que sin ustedes la culminación de este trabajo no hubiera sido posible.

Carolina Ramos R.

DEDICATORIA

Quiero dedicar este trabajo de disertación a mis padres Mery y José por ser quienes con su sacrificio me han permitido llegar hasta este punto, por enseñarme a perseguir mis metas, además de ser mis guías, mi fuente de inspiración y de amor.

Carolina Ramos R.

AGRADECIMIENTO

A todas las personas que día a día me brindaron su apoyo incondicional en estos años de preparación, a mis padres, hermana, Carolina, a mis profesores que me brindaron su apoyo en la realización del presente trabajo de disertación.

En especial a Ing. Alfredo Calderón quien me brindó su apoyo y su amistad en el trascurso de estos años.

Jonathan López

DEDICATORIA

Dedico este trabajo a mi familia, quienes me han apoyo a lo largo de mi vida brindándome su apoyo y enseñándome que la fortaleza, dedicación y perseverancia son los pilares fundamentales en la vida.

Jonathan López

Tabla de contenido

Prefacio 13
Antecedentes o Marco Referencial 15
JUSTIFICACIÓN 15
PROBLEMA 16
Planteamiento del Problema 16
OBJETIVOS 16
General 16
CAPÍTULO I: ¿QÚE ES BIG DATA? 17
1.1. Definición 17
1.1.1. Las 4 V's de Big Data 18
1.2. Historia/Evolución 19
1.3. Importancia 21
1.3.1. Predicción de eventos con Big Data 23
1.4. Conceptos Generales del Capítulo 23
1.4.1. Tabla de equivalencias de Informática 23
1.4.2. Datos estructurados, no estructurados y semi estructurados 24
1.4.3. Base de datos 25
1.4.4. Business Intelligence (Inteligencia de Negocio, BI) 25
CAPITULO II: Generalidades de Big Data 26
2.1. Áreas de Big Data 26
2.1.1. Recolección 26
2.1.2. Almacenamiento 27
2.1.3. Análisis 29
2.1.4. Visualización 32
2.2. Paradigmas de Big Data 32
2.2.1. MapReduce 33
2.2.2. Procesamiento Masivo en Paralelo (Massively Parallel Processing – MPP) 46
2.3. Conceptos Generales del Capítulo 46
2.3.1. Sistema de ficheros distribuidos. 46
2.3.2. Clúster 46
2.3.3. Cloud Computing 47
2.3.4. Bases de Datos 47
CAPITULO III: Herramientas y tecnologías de Big Data 50

3.1. Plataformas de Big Data 50
3.1.1. Apache Hadoop 50
3.1.2. Apache Spark 51
3.1.3. Oracle Big Data Appliance 52
3.2. Introducción a las Tecnologías de Big Data 54
3.2.1. Tecnologías Relacionadas con Apache Hadoop 55
3.2.2. Tecnologías Relacionadas Con Apache Spark 56
3.2.3. Tecnologías Relacionadas con Oracle Big Data Appliance 57
3.3. Tabla comparativa 57
CAPITULO IV: Instalación de un Ambiente de Big Data y Casos Prácticos. 59
4.1. Instalación de un Ambiente de Big Data 59
4.1.1. Selección de las herramientas 59
4.1.2. Requisitos de Instalación de Hadoop 60
4.1.3. Pasos de Instalación de Hadoop 61
4.2. Casos Prácticos 123
4.2.1. Ejecución Ejemplo WordCout (Contador de Palabras) 123
4.2.1.1. Pasos de Ejecución 124
4.2.2. Recolección de Tweets en una Base de Datos NoSQL. 139
CAPITULO V: Metodología de Proceso Enseñanza de Big Data. 163
Descripción de la Materia: 163
Objetivo General de la Materia: 163
5.1. Matriz de planificación didáctica acera de lo que es Big Data. 164
5.2. Matriz de planificación didáctica acera de las generalidades de Big Data. 165
5.3. Matriz de planificación didáctica acera de la enseñanza de las tecnologías y herramientas de Big Data. 166
5.4. Matriz de planificación didáctica acerca de la enseñanza de las tecnologías y herramientas de Big Data. 167
CAPITULO VI: Conclusiones y Recomendaciones 168
6.1. Conclusiones 168
6.2. Recomendaciones 170
Bibliografía 172
ANEXOS 176
ANEXO 1: Manual de Usuario Hadoop. 176
ANEXO 2: Manual de Usuario de Recolección de Tweets (CouchDB). 182

Índice de Figuras.

Figura 1.1: Las 4 V's de Big Data 19
Figura 1.2: Historia/Evolución Big Data. 21
Figura 1.3: Crecimiento de los datos a nivel mundial Fuente: (Shakuntala Gupta & Sabharwal, 2015) 21

Figura 2.1: Archivo de texto con datos estructurados acerca de los aeropuertos. 27
Figura 2.2: Visualización de datos en R. 30
Figura 2.3: Visualización de Atacames en ArcGis. 31
Figura 2.4: Visualización de los puntos recogidos. 31
Figura 2.5: Visualización de las áreas identificadas. 32
Figura 2.6: Esquema de un cálculo MapReduce. Fuente: (Leskovec, Rajaraman , & Ullman, 2014) 34
Figura 2.7: Esquemática de un cálculo MapReduce. Fuente: (Leskovec R. a.) 37
Figura 2.8: Visualización de Matriz M y Vector v 38
Figura 2.9: Matriz Dispersa. 40
Figura 2.10: Resultado Matriz C. 41
Figura 2.11: Representación de la Matriz como una lista de elementos no nulos. 41
Figura 2.12: Tarea "Map" de la multiplicación de la matriz. 42
Figura 2.13: Tarea "Reduce" de la Multiplicación de la matriz. 43
Figura 2.14: Tarea "Map" de la multiplicación de la matriz. 44
Figura 2.15: Resultado Final de la multiplicación de matrices. 45
Figura 2.16: Ejemplo de cubo de datos. Fuente: (Trujillo, Diseño y explotación de almacenes de datos: conceptos básicos de modelado multidimensional., 2013) 48
Figura 2.17: Ejemplo de tabla multidimensional. Fuente: (Trujillo, Diseño y explotación de almacenes de datos: conceptos básicos de modelado multidimensional., 2013) 49

Figura 3.1: Oracle Big Data Appliance Fuente: (Oracle, 2016) 54

Figura 4.1: Topología de un clúster de Hadoop en Modo Completamente Distribuido (multi-nodo). 60
Figura 4.2: Diagrama de Flujo del proceso de Instalación de Hadoop. 61
Figura 4.3: Jdk, Página web Oracle. 62
Figura 4.4: Acceso como root. 63
Figura 4.5: Copiar jdk a la ubicación root. 63
Figura 4.6: Instalación del jdk. 63
Figura 4.7: Instalación del jdk (proceso). 64
Figura 4.8: Definir variable JAVA_HOME. 64
Figura 4.9: Archivo para definir variable JAVA_HOME (1). 64
Figura 4.10: Archivo para definir variable JAVA_HOME (2). 65
Figura 4.11: Comandos necesarios para definir variable JAVA_HOME. 65
Figura 4.12: Comprobación Java es un directorio. 66
Figura 4.13: Abrir archivo Hosts 66
Figura 4.14: Archivo Hosts. 66
Figura 4.15: IP's y Dominios de los nodos del clúster. 67
Figura 4.16: Ping de comprobación de conexión de los nodos. 68

Figura 4.17: Página de descarga de Hadoop. 68
Figura 4.18: Copiar Hadoop al directorio root. 69
Figura 4.19: Descomprensión de Hadoop y Copia de Hadoop al directorio Hadoop. 69
Figura 4.20: Abrir fichero .bashrc 70
Figura 4.21: Configuración de variables de entorno de Hadoop en el fichero .bashrc. 71
Figura 4.22: Creación grupo Hadoop. 72
Figura 4.23: Creación de usuarios del grupo Hadoop. 73
Figura 4.24: Guardar script env-variable.sh en el equipo. 74
Figura 4.25: Guardar script directories_master.sh en el equipo. 74
Figura 4.26: Copiar scripts env-variable y directories_master al directorio root. 75
Figura 4.27: Permisos para ejecutar los scripts: env-variable y directories_master. 75
Figura 4. 28: Comprobación que los scripts se activaron correctamente. 75
Figura 4. 29: Ejecución de los scripts y creación de directorios. 76
Figura 4.30: Guardar script env-variable.sh en el equipo. 77
Figura 4.31: Guardar script directories_slaves.sh en el equipo. 78
Figura 4.32: Copiar scripts env-variable y directories_slaves al directorio root. 79
Figura 4.33: Permisos para Ejecutar los scripts: env-variable y directories_slaves. 79
Figura 4.34: Comprobación que los scripts se activaron correctamente. 79
Figura 4.35: Ejecución de los scripts y creación de directorios. 79
Figura 4.36: Abrir archivo core-site.xml en el Nodo Master. 82
Figura 4.37: Archivo core-site.xml Nodo Master. 83
Figura 4.38: Archivo core-site.xml Nodo Master. 83
Figura 4.39: Agregación de líneas de código al archivo core-site.xml Nodo Master. 84
Figura 4.40: Abrir archivo core-site.xml Nodos Esclavo. 85
Figura 4.41: Archivo core-site.xml Nodos Esclavo. 85
Figura 4.42: Archivo core-site.xml Nodos Esclavo. 86
Figura 4.43: Agregación de líneas de código al archivo core-site.xml Nodos Esclavo. 87
Figura 4.44: Copiar el Template mapred-site.xml al directorio donde se encuentra Hadoop. 88
Figura 4.45: Abrir el archivo mapred-site.xml Nodo Master. 88
Figura 4.46: Archivo mapred-site.xml Nodo Master. 88
Figura 4.47: Archivo mapred-site.xml Nodo Master. 89
Figura 4.48: Agregación de líneas de código al archivo mapred-site.xml Nodo Master. 90
Figura 4.49: Copiar el template del archivo mapred-site.xml al directorio donde se encuentra Hadoop. 90
Figura 4.50: Abrir archivo mapred-site.xml en los Nodos Esclavos. 90
Figura 4.51: Archivo mapred-site.xml en los Nodos Esclavos. 91
Figura 4. 52: Archivo mapred-site.xml en los Nodos Esclavos. 91
Figura 4.53: Agregación de líneas de código en el archivo mapred-site.xml en los Nodos Esclavos. 92
Figura 4.54: Abrir archivo hdfs-site.xml Nodo Master. 92
Figura 4.55: Archivo hdfs-site.xml Nodo Master. 93
Figura 4.56: Archivo hdfs-site.xml Nodo Master. 93
Figura 4.57: Agregación de líneas de código al archivo hdfs-site.xml Nodo Master. 94
Figura 4.58: Abrir archivo hdfs-site.xml Nodos Esclavo. 95
Figura 4.59: Archivo hdfs-site.xml Nodos Esclavo. 95
Figura 4.60: Agregación de líneas de código al archivo hdfs-site.xml Nodos Esclavo. 96

Figura 4.61: Abrir archivo yarn-site.xml Nodo Master. 97
Figura 4.62: Archivo yarn-site.xml Nodo Master. 97
Figura 4.63: Agregación de líneas de código al archivo yarn-site.xml Nodo Master. 98
Figura 4.64: Abrir archivo yarn-site.xml Nodos Esclavo. 99
Figura 4.65: Archivo yarn-site.xml Nodos Esclavo. 99
Figura 4.66: Agregación de líneas de código al archivo yarn-site.xml Nodos Esclavo. 101
Figura 4.67: Abrir fichero slaves para agregar los Nodos Esclavo. 102
Figura 4.68: Agregación de los Nodos Esclavo al fichero slaves. 102
Figura 4.69: Detener el funcionamiento del Firewall del sistema en el Nodo Master. 103
Figura 4.70: Abrir fichero sysctl.conf en el Nodo Master. 103
Figura 4.71: Desactivación del IPv6 en el Nodo Master. 103
Figura 4.72: Detener el funcionamiento del Firewall del sistema en los Nodos Esclavo. 104
Figura 4.73: Abrir fichero sysctl.conf en los Nodos Esclavo. 104
Figura 4.74: Desactivación del IPv6 en los Nodos Esclavo. 105
Figura 4.75: Ingreso como usuario hdfs. 105
Figura 4.76: Ingreso a la carpeta /bin del usuario hdfs. 105
Figura 4.77: Formato al Name Node. 106
Figura 4.78: Resultado de dar formato al Name Node. 106
Figura 4.79: Ir al directorio /sbin del usuario hdfs. 106
Figura 4.80: Inicio del demonio Name Node. 107
Figura 4.81: Inicio del demonio Secondary Name Node. 107
Figura 4.82: Usuario hdfs Nodos Esclavo. 107
Figura 4.83: Fichero /sbin usuario hdfs Nodos Esclavos. 107
Figura 4.84: Inicio del demonio Data Node. 108
Figura 4.85: Usuario yarn Nodo Master. 108
Figura 4.86: Fichero /sbin usuario yarn Nodo Master. 108
Figura 4.87: Inicio del demonio Resource Manager 108
Figura 4.88: Usuario yarn Nodos Esclavo. 109
Figura 4.89: Directorio /sbin del usuario yarn en los Nodos Esclavo 109
Figura 4.90: Inicio del demonio Node Manager. 109
Figura 4.91: Salir del usuario yarn Nodos Esclavo. 110
Figura 4.92: Verificación de los demonios activos en los Nodos Esclavo. 110
Figura 4.93: Salir del usuario yarn en el Nodo Master 110
Figura 4.94: Ingreso como usuario hdfs en Nodo Master. 110
Figura 4.95: Directorio /bin del usuario hdfs en Nodo Master 111
Figura 4.96: Creación de directorios user y temp en Nodo Master. 111
Figura 4.97: Verificación de los directorios creados. 111
Figura 4.98: Salir del usuario hdfs en el Nodo Master. 112
Figura 4.99: Sesión como usuario mapred en Nodo Master 112
Figura 4.100: Directorio /sbin del usuario mapred en Nodo Master 112
Figura 4.101: Inicio del demonio Job Histoy Server. 112
Figura 4.102: Directorio /bin del usuario mapred en Nodo Master 113
Figura 4.103: Ejecución del ejemplo pi que viene por defecto en Hadoop. 113
Figura 4.104: Resultado de la ejecución del ejemplo pi en Hadoop. 114
Figura 4.105: Resultado de la ejecución del ejemplo pi en Hadoop. 114
Figura 4.106: Resultado de la ejecución del ejemplo pi en Hadoop. 115

Figura 4.107: Comprobación de los componentes que se están ejecutando. 116
Figura 4.108: Vista en el explorador de los resultados del proceso ejecutado en Hadoop. 117
Figura 4.109: Vista en el explorador de los resultados del proceso ejecutado en Hadoop. 118
Figura 4.110: Vista en el explorador de la información de Hadoop. 119
Figura 4.111: Vista en el explorador de la información de Hadoop. 120
Figura 4.112: Rendimiento de la máquina antes de la ejecución de Hadoop. 121
Figura 4.113: Rendimiento de la máquina al inicio de la ejecución de Hadoop. 122
Figura 4.114: Rendimiento de la máquina durante la ejecución de Hadoop. 123
Figura 4.115: Eclipse, página web. 124
Figura 4.116: Copiar eclipse, a la dirección de usuario. 125
Figura 4.117: Verificar el archivo copiado. 125
Figura 4.118: Descomprimir el archivo. 125
Figura 4.119: Enlace a directorio /bin. 125
Figura 4.120: Editar lanzador de Gnome. 126
Figura 4.121: Editar lanzador de Gnome con gestor vi. 127
Figura 4.122: Resultado del lanzador. 127
Figura 4.123: Directorio de Eclipse. 128
Figura 4.124: Selección de Java Project. 129
Figura 4.125: Nombre de la aplicación. 129
Figura 4.126: Creación de la clase. 130
Figura 4.127: Importar librerías. 131
Figura 4.128: Importar librerías jar externas. 132
Figura 4.129: Destino de Exportación JAR file. 134
Figura 4.130: Guardar .jar en la dirección establecida. 135
Figura 4.131: Archivo de texto. 135
Figura 4.132: Verificar el almacenamiento de los archivos. 136
Figura 4.133: Acceder a la carpeta que contiene archivos 136
Figura 4.134: Copiar archivo a sistema hdfs. 136
Figura 4.135: Consulta del fichero con permisos -rw-r--r--. 137
Figura 4.136: Ejecución en Hadoop con archivos de texto. 137
Figura 4.137: Resultado de operación. 138
Figura 4.138: Resultado de operación 2. 138
Figura 4.139: Resultado final Word Count. 139
Figura 4.140: Estructura de Tweets. 140
Figura 4.141: Diagrama de flujo del proceso de recolección de Tweets. 142
Figura 4.142: Instalación repositorio actual de CouchDB. 143
Figura 4.143: Actualización de los paquetes. 143
Figura 4.144: Eliminar binarios de CouchDB. 144
Figura 4.145: Instalar CouchDB. 144
Figura 4.146: Para CouchDB 144
Figura 4.147: Arrancar servicio de CouchDB. 145
Figura 4.148: Verificar si se encuentra encendido CouchDB. 145
Figura 4.149: Acceso a CouchDB y a sus funciones. 146
Figura 4.150: Instalar tweepy. 147
Figura 4.151: Descargar librería CouchDB-09. 147
Figura 4.152: Descomprimir librería. 148

Figura 4. 153: Acceso a carpeta CouchDB-09. 148
Figura 4.154: Instalación de librerías para que puedan ser reconocidas por python. 149
Figura 4.155: Importación de CouchDB en python. 149
Figura 4.156: Importación tweepy. 150
Figura 4.157: Twitter Application Management. 150
Figura 4.158: Acceso a Application Management. 151
Figura 4.159: Creación de la Aplicación. 152
Figura 4.160: Claves generadas por Twitter. 153
Figura 4.161: Claves generadas por Twitter 2. 154
Figura 4.162: Claves como Consumer Key, y Consumer Secret. 155
Figura 4.163: Interfaz de CouchDB, creación de nueva base de datos. 155
Figura 4.164: Creación de base de datos denominada quito. 156
Figura 4.165: Verificación de base de datos. 156
Figura 4.166: Código fuente desarrollado en Python, llamado tweeps.py. 157
Figura 4.167: Boundingbox, presentación de coordenas. 158
Figura 4.168: Acceso a la carpeta que contiene el código. 158
Figura 4.169: Ejecutar tweets.py. 159
Figura 4.170: Base de datos con alrededor de 15421 Tweets. 160
Figura 4.171: Ilustración del almacenamiento. 160
Figura 4.172: Información tweet registrado. 161
Figura 4.173: País donde fue emitido el tweet. 162

Índice de Tablas

Tabla 1.1: Tabla De Unidades Básicas De Información Y Tratamiento De Datos. Fuente: (Jiménez, Big Data. Un nuevo paradigma, 2014) Elaboración: López / Ramos 24
Tabla 2.1: Tipos de datos en el paradigma Big data. Fuente: (Jiménez, 2014) 34
Tabla 3.1: Costos de infraestructura y servicio Premier Support. Fuente: (Jean-Pierre Dijcks-Oracle, 2014) 52
Tabla 3.2: Elementos de una implementación de grandes volúmenes de datos Mediana Empresa de costo / beneficio. Fuente: (Nik Rouda, Senior Analyst and Adam DeMattia, Research Analyst, 2015) 53
Tabla 3.3: Tabla comparativa Apache Spark, Apache Hadoop, Oracle Big Data Appliance. 57

Prefacio

La elaboración de la presente disertación de grado surgió de la necesidad de profundizar en un tema actual y de gran relevancia en el campo de la informática, es por eso que al analizar distintas tecnologías se encontró un tema vigente, innovador y prometedor como lo es “Big Data”.

Durante el transcurso de esta carrera se pudo conocer la importancia que genera el poder procesar y analizar correctamente los datos, así como las desventajas que acarrean el no hacerlo. Se sabe que con las nuevas herramientas tecnológicas los datos cada vez son más voluminosos y se transmiten más rápidamente por lo cual ahondar en este tema es algo atractivo. De esta manera, se ha vuelto un reto estudiar todo lo concerniente a “Big Data” con el fin de obtener la mayor información posible que permita la elaboración de una “Guía Metodológica” acerca del presente tema.

La idea de elaborar esta Guía Metodológica para el Proceso Enseñanza de Big Data, surgió por la importancia que está teniendo este tema en el mundo de hoy. Todas las empresas, compañías, organizaciones, etc. quisieran optimizar y procesar la gran cantidad de datos que reciben diariamente para poder obtener información que genere valor y tomar correctamente sus decisiones. Además, que la mayoría de información que se encuentra en los libros está en otro idioma y muchas veces dicha información está estructurada de una manera técnica lo cual dificulta el fácil entendimiento y estudio de “Big Data”.

Por otro lado, hoy en día con el auge y la importancia que están teniendo las redes sociales tales como Facebook, Twitter, Linkedin, etc., representan una de las fuentes más significativas para obtener datos masivos, lo cual da lugar a una línea de investigación importante, ya que gracias a los datos recogidos a través de estos medios de comunicación, muchas empresas pueden verse beneficiadas al permitirles tener una observación inmediata del comportamiento de sus consumidores sin que éstos se sientan observados, por ejemplo Netflix al utilizar Big Data puede predecir lo que más le gustará a su audiencia y recomendarles contenido, que por cierto, su herramienta de recomendación es una de las cosas que más gusta a sus usuarios. Esto es hacer Big Data, el poder integrar toda esa información y obtener datos operativos.

Al mismo tiempo, Big Data está presente en los diferentes campos de las ciencias como por ejemplo en la medicina que es donde se generan una gran cantidad de datos, muchos de

ellos datos no estructurados, provenientes de formatos escritos en papel como recetas médicas o electrónicos que permanecen sin utilizar por la dificultad y la falta de disponibilidad de herramientas que permitan gestionarlos de manera efectiva. Pero actualmente, gracias a la integración de Big Data en el campo de la salud esos datos podrán ser procesados de una forma más eficiente, de tal manera que Big Data se utilizará para predecir y prevenir con mayor certeza los diferentes tipos de enfermedades. Así se puede evidenciar como Big Data es importante no solo en la medicina, sino también en la economía, las finanzas, los negocios, etc.

Es entonces que por estas razones se eligió elaborar esta guía en donde se hablará todo lo referente a Big Data y a la vez, se demostrará con casos prácticos lo que es hacer Big Data en nuestros tiempos.

Antecedentes o Marco Referencial

En la actualidad el volumen de los datos ha crecido de manera exponencial e inmensurable, lo que ha provocado el difícil manejo de los mismos, causando un problema en la comprensión y toma de decisiones; para la optimización de esta situación, se desarrolló lo que ahora se conoce como Big Data, pero teniendo en cuenta la escasez de información, libros en un idioma distinto al español, falta de difusión del tema, provoca que las personas tengan poca accesibilidad en la adquisición de este conocimiento.

JUSTIFICACIÓN

Relevancia Social: Actualmente, el auge de redes sociales, el crecimiento de la población, entre otros factores, han provocado el crecimiento vertiginoso y voluminoso de los dantos, ante lo cual, las empresas han tenido que enfrentarse a nuevos retos que les permitan descubrir, analizar y sobretodo, entender el funcionamiento de herramientas no tan tradicionales. Es así, que resulta atractivo e interesante profundizar en este tema y proponer una guía metodológica hacia lo que es Big Data, lo que significa y su importancia actualmente.

Relevancia Académica: Generar una guía metodológica para el proceso enseñanza-aprendizaje de Big Data es un reto innovador puesto que es un tema ya establecido pero que necesita ser profundizado y sobretodo debe quedar claro en personas afines a carreras semejantes a Ingeniería en Sistemas de Información. Además, se enfocará en casos prácticos, tales como; instalación y configuración de un servidor para Big data, ejercicios prácticos que vinculen la teoría a casos funcionales. Esto puede servir de apoyo a profesores interesados en dictar clases acerca de Big Data y, al mismo tiempo ser útil para facilitar el aprendizaje de los estudiantes.

Relevancia Personal: La elaboración de una guía metodológica para el proceso enseñanza-aprendizaje de Big Data proporcionará una ventaja en el desarrollo profesional, debido a que se podrá conocer a fondo de qué se trata este tema que en la actualidad está tomando un peso relevante en el manejo de la información y al mismo tiempo, permitirá ampliar el conocimiento personal y por lo tanto, facilitará el desenvolvimiento en el desarrollo de aplicaciones o herramientas concernientes a este tema.

PROBLEMA

Planteamiento del Problema

Con el transcurso del tiempo se generan volúmenes masivos de datos, lo que se conoce como Big Data. Estos datos pueden ser irrelevantes para las personas e incluso pueden llegar a ocupar grandes cantidades de almacenamiento, debido a que Big Data es la herramienta que se encarga de la optimización de recursos y del manejo de datos; además, la falta de existencia de libros relacionados a este tema, información en inglés, costos elevados, resulta muy difícil tanto para las empresas, profesores, estudiantes acceder y aprovechar este conocimiento. Por lo que es necesario establecer una guía metodológica que facilite el aprendizaje incluyendo casos prácticos, tales como; instalación y configuración de un servidor para Big data (Hadoop), ejercicios prácticos que vinculen la teoría a casos funcionales.

OBJETIVOS

General

Elaborar una guía metodológica para el proceso enseñanza- aprendizaje de Big Data que beneficie a los docentes, además realizar la instalación y configuración de una plataforma tecnológica de Big Data.

Específicos

- Establecer una metodología innovadora para el proceso de enseñanza aprendizaje de Big Data.
- Fundamentar teóricamente las bases de Big data.
- Determinar cuál es la importancia de utilizar Big Data y qué información se puede considerar más relevante para el proceso de enseñanza aprendizaje.
- Investigar cuáles son los métodos existentes e innovadores que permitan la descomposición de información
- Analizar las plataformas que permiten hacer Big data.

CAPÍTULO I: ¿QÚE ES BIG DATA?

El objetivo de este capítulo es exponer la función de Big data desde su historia/ evolución, su importancia con su respectiva relevancia actual. Con la finalidad que más adelante sea mucho más fácil comprender los términos realizados en esta guía.

1.1. Definición

A simple vista el término Big Data se lo puede entender como algo que es grande y está lleno de información, pero esta definición no proporciona algo claro, ni describe lo que Big Data es en realidad. Entonces, ¿Qué es exactamente Big Data?, Big Data es un término muy conocido en el medio actual y es a menudo descrito como conjuntos de datos que son extremadamente grandes, es decir, que han crecido más allá de la capacidad para poder ser gestionados y analizados con herramientas tradicionales de procesamiento de datos. En otras palabras, los datos han crecido de manera gigante, lo cual hace dificultoso su manejo y aún más difícil el poder extraer valor de ellos al presentarse ciertas dificultades como la adquisición, el almacenamiento, la búsqueda y el intercambio.

Conforme un estudio realizado por Cisco, “entre el 2015 y el 2020 el tráfico de datos móviles que habrá en el 2020 será de 5,5 millones de usuarios móviles en el mundo, frente a los 4,8 mil millones en 2015. Además, En 2020, habrá 11,6 millones de dispositivos y conexiones móviles listos, casi 4 mil millones más que en 2015.
Asimismo, en el año 2020, la velocidad de conexión móvil media aumentará 3,2 veces, pasando de 2,0 Mbps en 2015 a 6,5 Mbps para el año 2020. Del mismo modo, el tráfico IP móvil global alcanzará una tasa anual de 367 exabytes en 2020, frente a los 44 exabytes en 2015.” (Cisco, s.f.)

Para poder entender mejor este concepto de Big Data se encontraron otras definiciones formales que pueden ser de gran ayuda para esclarecerlo, como por ejemplo las que están planteadas a continuación.

“Big data es un término usado para describir los datos que tienen volumen masivo, vienen en una variedad de estructuras, y se generan a alta velocidad. Este tipo de datos plantean desafíos a los sistemas tradicionales de bases de datos que se utilizan para almacenarlos y

procesarlos. Los Grandes Datos están abriendo camino para nuevos enfoques de procesamiento y almacenamiento de datos". (Shakuntala Gupta & Sabharwal, 2015).

"En términos generales Big Data se puede referir a la tendencia en el avance de la tecnología que ha abierto las puertas hacia un nuevo enfoque de entendimiento y toma de decisiones, la cual es utilizada para describir enormes cantidades de datos **(estructurados, no estructurados y semi estructurados)**[1] que tomaría demasiado tiempo y sería muy costoso cargarlos a una base de datos relacional para su análisis. De tal manera que, el concepto de Big Data aplica para toda aquella información que no puede ser procesada o analizada utilizando procesos o herramientas tradicionales. Sin embargo, Big Data no se refiere a alguna cantidad en específico, ya que es usualmente utilizado cuando se habla en términos de petabytes y exabytes de datos". (Ricardo Barranco Fragoso, IT Specialist for information Management, IBM Software Group , 2012)

1.1.1. Las 4 V's de Big Data

En cuanto a la extracción de valor, Big Data tiene distintivos importantes que permiten la revelación de la información, estos distintivos consienten maximizar el valor respecto al negocio o cualquier aplicación que sea desarrollada. Como líder en el sector, los analistas de datos de IBM clasifican a los datos en cuatro dimensiones: volumen, velocidad, variedad y veracidad.

Volumen: Trata acerca del gran tamaño de los datos. Las empresas están inundadas de datos, acumulando fácilmente terabytes e incluso petabytes de información. Hoy en día, los datos no solo se generan dentro de una empresa si no también se generan en base a transacciones, la cantidad de tráfico de datos móviles crecerá de manera inmensurable.

Variedad: Big Data se extiende más allá de los datos estructurados para incluir datos no estructurados de todas las variedades, estos datos son generados a partir de diversos dispositivos móviles y fuentes que no siguen una estructura fija tal como: archivos de texto, videos, fotos, PDF, audio y otros formatos no estructurados.

Veracidad: Las enormes cantidades de datos recogidos pueden conducir a errores estadísticos y mala interpretación de la información recolectada. La pureza de la información es fundamental para el valor.

[1] Revisar el Subcapítulo 1.5.2. : Datos estructurados, no estructurados y semi estructurados

Velocidad: Es el acceso y el flujo de grandes datos, necesarios para ser analizados. Estos datos son susceptibles en el tiempo. Si los datos no pueden ser procesados a la velocidad requerida, pierden su significado.

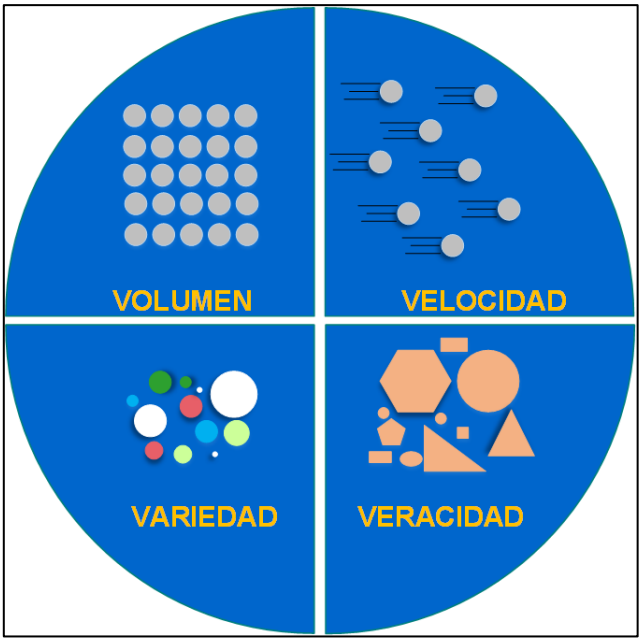


Figura 1.1: Las 4 V's de Big Data

1.2. Historia/Evolución

Teniendo en cuenta que el ser humano desde tiempos inmemorables no ha dejado de recopilar información. La ideología de Big Data probablemente puede ser remontada a los días antes de la era de las computadoras. Cuando los datos no estructurados, registros en papel, eran más frecuentes, en vista que no existían herramientas que permitían la gestión de los datos.

Fue entonces que en el año de 1880 se presentó el primer desafío de Big Data con el censo de Estados Unidos, el cual arrojó una montaña de datos de aproximadamente 50 millones de personas, y solo la tecnología limitada estaba disponible para hacer algún tipo de análisis. Así fue como este problema de Big Data (de los grandes datos) no pudo ser resuelto durante este año, por lo que se necesitaron siete años para tabular y presentar información de los datos recogidos de manera manual.

Pero ya para el año de 1890 las cosas empezaron a cambiar gracias a la introducción de la primera plataforma de Big Data, que consistía en un dispositivo mecánico llamado Sistema

de tabulación de Hollerith, que trabajaba utilizando tarjetas perforadas y podía contener cerca de 80 variables. Este sistema, revolucionó el valor de los datos del censo y su análisis tardó seis semanas, en lugar de siete años.

A continuación, en el año de 1940 **Turing y Good**[2] realizaron un trabajo distintivo para decodificar los mensajes alemanes en la segunda guerra mundial. Pero el siguiente gran salto para el análisis de Big Data se dio en el año de 1944 con el Proyecto de Manhattan, el mismo que consistía en un proyecto científico que ocurrió durante la Segunda Guerra Mundial por parte de los Estados Unidos con la colaboración parcial de Reino Unido y Canadá, cuyo objetivo primordial era desarrollar la primera bomba atómica antes que la Alemania nazi lo lograra. El equipo de este proyecto realizó simulaciones por ordenador para predecir el comportamiento de una reacción nuclear en cadena.

Así muchos cambios siguieron apareciendo, y en el año de 1966 **SAS Institute**[3] comenzó como un proyecto de investigación financiado por el Ministerio de Agricultura de Estados Unidos. Posteriormente, en "1973 se crea el modelo Black-Sholes para predecir el precio óptimo de las acciones en el futuro". (FICO, 2013)

Conforme pasaron los años la tecnología fue induciendo cambios impactantes, es así como Amazon y Ebay aparecieron en el año de 1995, comenzando de esta manera la personalización de la experiencia online. Seguidamente, en "1998 Google aplica algoritmos a las búsquedas web, para maximizar la relevancia de los resultados". (FICO, 2013)

Finalmente ya entrando a nuestra era, es decir, desde el año 2000 hasta el presente, este concepto de Big Data se reafirma con el surgimiento de redes sociales, teléfonos inteligentes, dispositivos electrónicos, la nube, etc., ante lo cual los analistas mencionan que hoy en día se generan 2,5 trillones de bytes relacionados con el término de Big Data.

[2] **Alan Turing:** (Alan Mathison Turing; Londres, 1912-Wilmslow, Reino Unido, 1954) Matemático británico. En la Segunda Guerra Mundial ofreció un marco de aplicación práctica de sus teorías, consistía en descifrar los mensajes codificados que la Marina alemana empleaba para enviar instrucciones a los submarinos que hostigaban las filas de ayuda material enviados desde Estados Unidos. (Biografías y vidas, 2004)
Good: (Irving Ioannes Good; 9 de Diciembre de 1916, 5 de abril de 2009). Matemático británico. Participó como criptógrafo en descifrar los mensajes en la segunda Guerra Mundial. (Los Angeles Times, 2009)

[3] SAS: Es una empresa de software de capital privado más grande del mundo líder en Business Analytics, fundada en Estados Unidos. Ofrece un verdadero poder analítico a empresas que manejan altos volúmenes de datos y destilan la información esencial, para facilitar el proceso de toma de decisiones utilizando modelos predictivos y descriptivos, pronósticos, simulación y optimización. (SAS México, 2014)

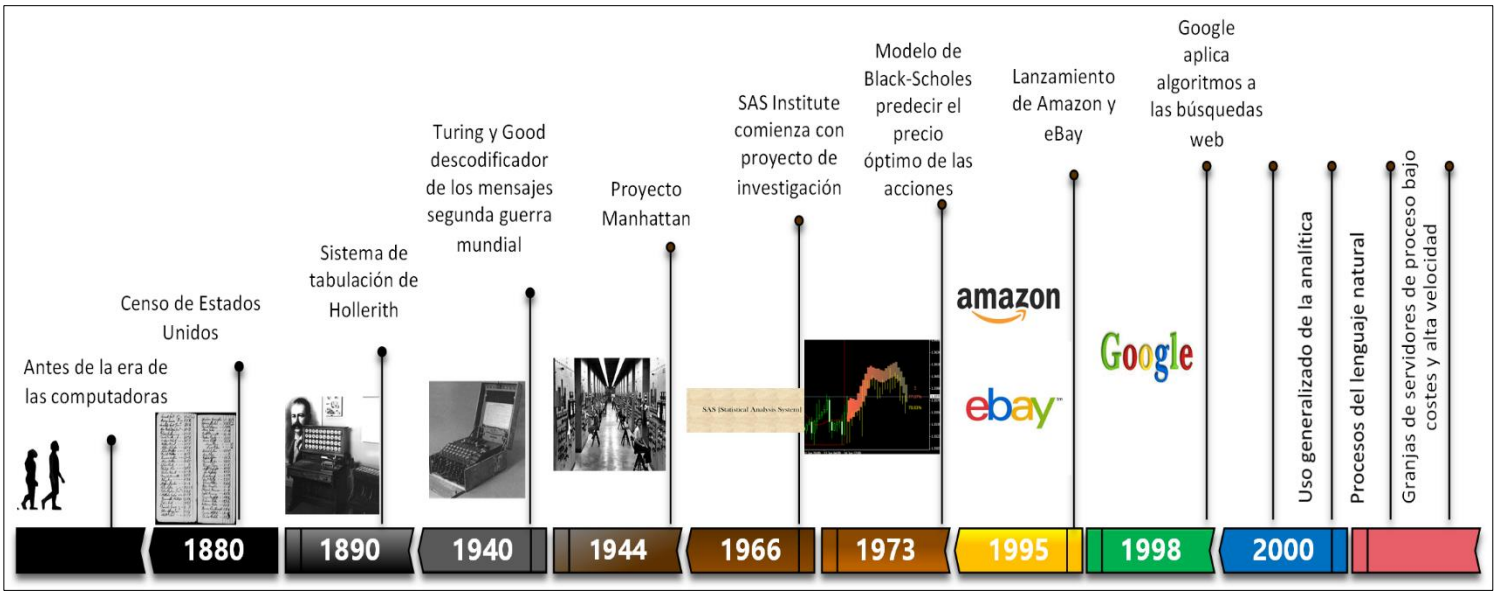


Figura 1.2: Historia/Evolución Big Data.

1.3. Importancia

Big Data, junto con la nube, redes sociales, análisis y movilidad, son palabras de moda hoy en día en el mundo de la informática. La disponibilidad de Internet y dispositivos electrónicos para las masas está aumentando cada día. Específicamente, smartphones, redes sociales y otros dispositivos como tablets y sensores están creando una explosión de datos. Los datos se generan a partir de diversas fuentes en diversos formatos como el vídeo, texto, voz, archivos de registro, y las imágenes. Un solo segundo de video de alta definición (HD) genera 2.000 veces más bytes que la de una sola página de texto.

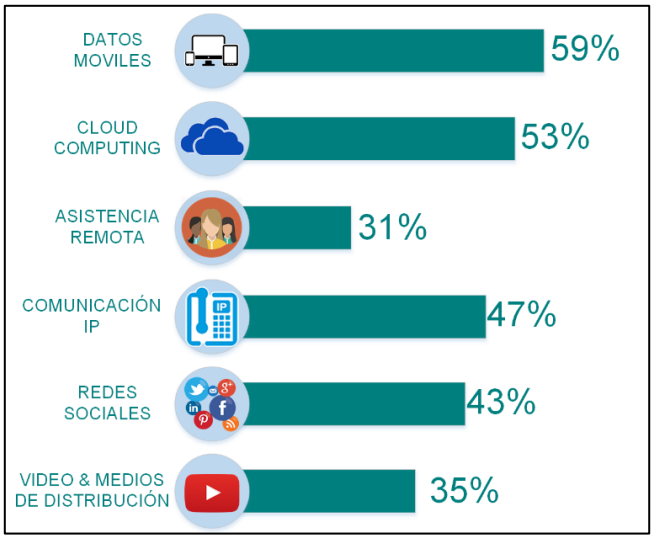


Figura 1.3: Crecimiento de los datos a nivel mundial Fuente: (Shakuntala Gupta & Sabharwal, 2015)

Como se puede observar con la Figura 1.3, la información crece de manera veraz, por lo cual, estos datos en gran cantidad cambian la ejecución de procesos de negocio y las organizaciones se favorecen al indagar en el crecimiento de los datos. Puesto que el tener técnicas de recopilación y análisis de datos, beneficia al sector de los negocios para que puedan tener una visión más clara de las preferencias y gustos de sus clientes. Entonces, se puede decir que, el análisis de datos en las organizaciones promueve productividad, crecimiento y beneficios tanto al productor como al consumidor.
Las organizaciones están invirtiendo recursos en nuevas tecnologías orientándose cada vez a la acumulación de datos y análisis.

A continuación se explican unos ejemplos que evidencian casos claros acerca de “Big Data”:

Cuando empresas como Amazon recomiendan qué libro leer, lo que hay detrás del proceso es “Big Data”. Las compañías aplican algoritmos cuando combinan datos como los gustos de un usuario al que conoce por adquisiciones anteriores u otros que lo identifican como sus visitas a determinadas páginas web.

De acuerdo con (Marr, 2015), “algunas empresas aprovechan los datos en grandes cantidades para impulsar el rendimiento del negocio. Entre estas empresas se encuentran, Google, Amazon, Facebook, General Electric y Microsoft. Entre las primeras empresas se encuentra Google, la cual procesa 3.5 millones de solicitudes por día, y cada solicitud consulta a una base datos de 20 millones de páginas web.”, por otro lado Google es uno de los negocios más rentables en el mundo, cuenta con gran cantidad de información y además utiliza PageRank que es un conjunto de algoritmos indexados por un motor de búsqueda. Su función es avanzar más lejos en las búsquedas basadas en palabras claves, hacia búsquedas semánticas. Esto implica analizar no sólo las palabras en la consulta si no también la conexión entre ellas, para determinar la mayor importancia que una página web tiene en Internet. Por ejemplo, “Google se hace la idea de que cuando una página coloca un enlace (link) a otra, es de hecho un voto para esta última.

Cuantos más votos tenga una página, será considerada más importante por Google. Además, la importancia de la página que emite su voto también determina el peso de este voto. De esta manera, Google calcula la importancia de una página gracias a todos los votos que reciba, teniendo en cuenta también la importancia de cada página que emite el voto”. (Mipagerank, 2003)

Sin embargo, después de todo, las búsquedas son gratis, Entonces, ¿Cómo Google ha logrado ser un negocio rentable en el planeta? Google acumula grandes cantidades de datos

sobre las personas que lo usan, mientas las empresas pagan generosamente a Google, para que los anuncios aparezcan en los ordenadores de sus clientes.

1.3.1. Predicción de eventos con Big Data

"Aprovechar el poder de la ciencia de datos en el servicio de la humanidad." (Datakind, 2015)

Con gran variedad de datos actuales e históricos, se puede realizar predicciones sobre futuros eventos. Las predicciones no suelen ser afirmaciones absolutas, llegan a una cierta probabilidad de que algo suceda.

Por ejemplo, la predicción de zonas geográficas vulnerables a crímenes. Hoy en día se han creado aplicaciones reales que permiten predecir zonas en las cuales se pueden producir crímenes a partir de un perfil sociodemográfico de la zona, obteniendo como resultado las zonas en que se cometerá un delito.

Otra aplicación que resulta muy útil es, la detección de epidemias de gripe a partir del motor de búsqueda. Google asegura que puede rastrear enfermedades similares a la influenza en una población, este método consiste en determinar la frecuencia de consultas altamente correlacionadas con el porcentaje de visitas al médico que un paciente presenta con síntomas similares a la gripe, Google afirma que se puede estimar con precisión el nivel de actividad de la influenza semanalmente en cada región de los Estados unidos.
Como se puede observar en los ejemplos ya mencionados con anterioridad, Big Data en los últimos años, ha logrado posesionarse tanto en el mundo de los negocios como en muchos aspectos de la vida cotidiana, al convertirse en una herramienta indefectible para las ventajas competitivas de grandes y medianas empresas, al permitir transformar los datos en grandes resultados a través del análisis de los mismos. Al mismo tiempo, se esperan grandes avances en el futuro en todo lo que concierne a Big Data.

1.4. Conceptos Generales del Capítulo

1.4.1. Tabla de equivalencias de Informática

Tabla 1.1: Tabla De Unidades Básicas De Información Y Tratamiento De Datos. Fuente: (Jiménez, Big Data. Un nuevo paradigma, 2014) Elaboración: López / Ramos

Tabla De Unidades Básicas De Información Y Tratamiento De Datos				
Nombre	Símbolo	Sistema internacional	Ejemplo 2014 estático	Ejemplo 2014 dinámico
Byte	B	10^0 bytes	1 B es un número de 0 a 255	
Kilobyte	KB	10^3 bytes	2 KB es aproximadamente un sector de CD-ROM	
Megabyte	MB	10^6 bytes	3 MB es aproximadamente una canción de 3 minutos	4 MB/min en llamadas de vídeo por Skype
Gigabyte	GB	10^9 bytes	8/16 GB es el tamaño estándar de mercado de un pen-drive	4 GB/hora de vídeo de alta calidad
Terabyte	TB	10^{12} bytes	4 TB es el tamaño de un disco de 120 € que almacena 800.000 fotos o canciones mp3	20 TB/hora es la información generada por un motor de avión en el aire
Petabyte	PB	10^{15} bytes	2 PB es la información almacenada en todas las bibliotecas de investigación académicas de USA	24 PB/día es la información recogida por Google
Exabyte	EB	10^{18} bytes	5 EB es aproximadamente todas las palabras pronunciadas por todos los seres humanos	966 EB es aproximadamente la predicción del volumen total de Internet en 2015
Zetabyte	ZB	10^{21} bytes	Se estimó que en 2012 la capacidad instalada de almacenamiento de información en el mundo sería de 2,5 ZB.	5 ZB/año es la cantidad de datos digitales promedio que se van a generar en la Tierra en los próximos 8 años
Yottabyte	YB	10^{24} bytes	1 YB equivale a la capacidad del Data Center inaugurado por la NASA en 2013	
Xerabyte	XB	10^{27} bytes	1 XB equivale a 1.257.000 iPad 3 de máxima capacidad por cada habitante de la tierra	

1.4.2. Datos estructurados, no estructurados y semi estructurados

Dato

Como se puede observar se habla mucho acerca de los datos, entonces ¿Qué es un dato? Un dato es el componente fundamental de las bases de datos, están relacionados entre sí formando un conjunto con mínimas redundancias. Los datos por sí mismos no aportan conocimiento hay que procesarlos y transformarlos para obtener información. (Cobo, 2009)

Existen distintos tipos de datos que de acuerdo a su clasificación proporcionan perspectivas diferentes.

Datos estructurados: "Son el resultado de tomar datos organizados y formatearlos para facilitar el almacenamiento, uso y generación de información." (Carlos Coronel, 2011)

Datos semi estructurados: "Son datos que ya han sido procesados en alguna medida. Por ejemplo, si observamos una página web cualquiera, los datos se presentan en formato arreglado previamente para presentar alguna información." (Carlos Coronel, 2011)

Datos no estructurados: "Se refiere típicamente a aquellos datos que no están organizados bajo el Modelo de Datos Relacional, definido por Edgar Codd en 1970. Algunos ejemplos comunes de información no estructurada son los archivos de texto, documentos (P DF, Word), imágenes, audio y video, entre otros." (Alexander Ambriz Rivas, M. Sc., Client Technical Professional, 2013)

1.4.3. Base de datos

Una base de datos es un conjunto de datos almacenados sin redundancias innecesarias en un soporte informático y accesible simultáneamente por distintos usuarios y aplicaciones. Los datos deben estar estructurados y almacenados de forma totalmente independiente de las aplicaciones que la utilizan. (Cobo, 2009)

1.4.4. Business Intelligence (Inteligencia de Negocio, BI)

Consta de una categoría amplia de aplicaciones y tecnologías para recopilar, almacenar, analizar, y facilitar el acceso a los datos. BI ofrece información útil, que ayuda a los usuarios de las empresas a tomar mejores decisiones de negocio utilizando sistemas de apoyo basados en hechos. BI funciona mediante el uso de un análisis en profundidad de los datos comerciales detallados, proporcionados por bases de datos, datos de la aplicación, y otras fuentes de datos tangibles. En algunos círculos, BI puede proporcionar vistas históricas, actuales y predictivas de las operaciones comerciales. (Ohlhorst F. J., 2012)

CAPITULO II: Generalidades de Big Data

El objetivo de este capítulo es profundizar temas relevantes acerca de Big Data, es decir, conceptos que son muy utilizados hoy en día y que son de gran importancia para entender cómo funciona y en qué consiste hacer Big Data.

2.1. Áreas de Big Data

2.1.1. Recolección

Cuando se habla de recolección de datos se habla de una de las disciplinas que ha variado con mayor rapidez y en la menor cantidad de tiempo, debido a que los datos se generan en cifras millonarias y provienen de una gran variedad de dispositivos existentes en todo el mundo que emiten, procesan y recogen datos de diferentes actividades como la información existente en redes sociales, en la nube, variables de geolocalización, entre otros.

Actualmente, muchas empresas están logrando especializarse en la recolección de datos a través de diversos programas que "añaden además algo de inteligencia y consiguen que los números recolectados sean coherentes y homogéneos para poder así realizar mejor los procesos posteriores". (Tablet Army, 2012)

Este ejemplo pretende ilustrar el concepto mencionado, para ello se utilizará como referencia una herramienta llamada OpenFlights (http://openflights.org/data.html), que permite determinar vuelos en todo el mundo, buscar y filtrar todo tipo de información, el uso de esta página web es gratuita. Además cuenta con variedad de módulos, donde presenta información de aeropuertos, información de aerolíneas y las informaciones de rutas. Para este ejemplo, se ha tomado información correspondiente a los distintos aeropuertos en el mundo.

Según OpenFlights, el mundo hasta el 2009 contaba con 6977 aeropuertos. Por lo tanto, se tomará esta información y se la utilizará en las siguientes áreas de Big Data.

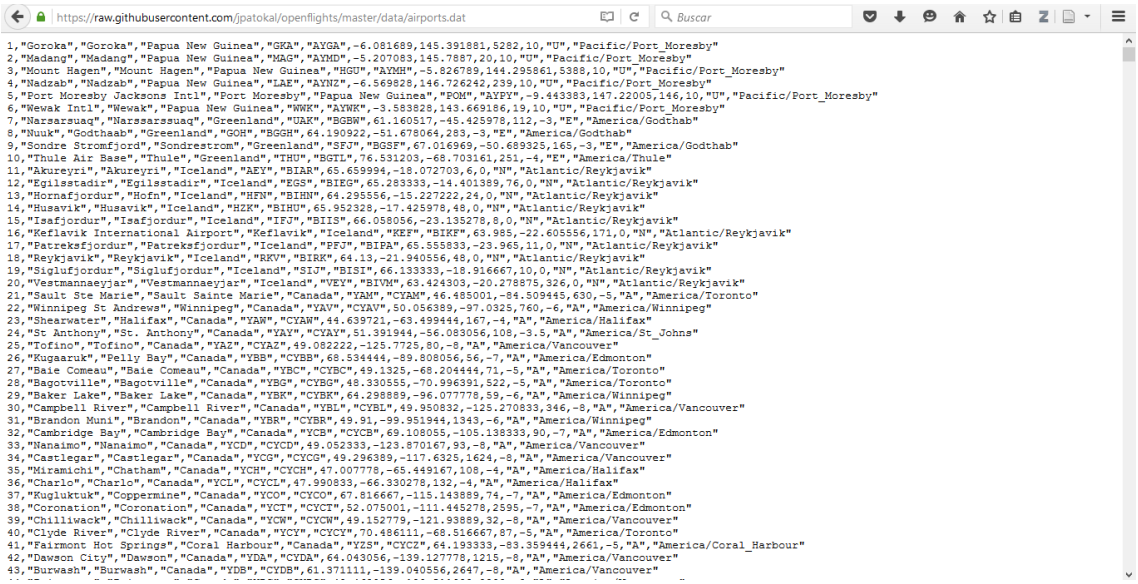


Figura 2.1: Archivo de texto con datos estructurados acerca de los aeropuertos.

El archivo contiene 9541 registros, con campos estructurados donde se detalla aeropuerto ID, nombre de la ciudad, País, Asignación al país sea dentro o fuera del país, código OACI, latitud, longitud, altitud, zona horaria.

La información obtenida representa un gran volumen de datos, es estructurada, pero no tiene todas las características de Big Data.

2.1.2. Almacenamiento

De acuerdo con (Tablet Army, 2012), las empresas actualmente tienen archivados sus datos, pero no saben cómo procesarlos. Ante la gran cantidad de datos que se generan diariamente, se debe llevar a cabo un almacenamiento escalable, es decir, un sistema que pueda variar su tamaño de almacenamiento (ya sea aumentándolo o disminuyéndolo) según las necesidades, y que esto no afecte al rendimiento general de todo el sistema.

A partir de esta necesidad aparecieron los **sistemas de ficheros distribuidos**[4], que consisten en la distribución y el compartimiento de cierta información por las redes. También

[4] Revisar el Subcapítulo 2.3.1: Sistema de ficheros distribuidos.

los **clúster**[5] de ordenadores o nodos interconectados, que se encuentran dispuestos de esta manera para tener un sólo sistema de ficheros lógicos.

Por otro lado, el almacenamiento se está trabajando también en la "nube", está ayudando a la consolidación del nuevo mundo del Big Data. "La multinacional norteamericana EMC, por ejemplo, sugiere optar por infraestructuras orientadas directamente a Big Data, a través de un almacenamiento que incluya tecnología **Cloud**[6]**.**" (Tablet Army, 2012). Resulta oportuno mencionar, que se puede contar con un repositorio de más de 10 petabytes en la nube, en lugar de islas de datos, que por lo general, suelen requerir una administración manual y además, están soportadas en sistemas dispares.

De acuerdo con los razonamientos que se han venido realizando es oportuno preguntar, ¿Qué tipo de datos almacena Big data? Big Data permite el almacenamiento de datos estructurados y no estructurados y semi estructurados. Por esta razón, las **bases de datos relacionales**[7]**,** que están basadas en lenguaje SQL (Structured Query Language), no pueden manejar datos no estructurados y semi estructurados. Además, son bases de datos poco flexibles ya que cuando se crea su estructura es bastante conflictivo realizar cambios en ésta (como añadir nuevas columnas a una tabla o cambiar el tipo de una columna).

Debido a esta complejidad existen las llamadas **bases de datos NoSQL**[8] (Not only SQL): las empresas que basan su actividad en Internet y las redes sociales tales como Google, Facebook, Amazon, Twitter. Son empresas que no siguen el modelo de base de datos relacionales, y aportan más flexibilidad al no requerir estructuras fijas como tablas. Algunos ejemplos de sistemas NoSQL son MongoDB, Hadoop. Los sistemas NoSQL acostumbran a ser en este sentido más adaptables a los sistemas distribuidos, permiten una mayor flexibilidad en la configuración de las maquinas con hardware más modesto. (Morros, 2013)

Continuando con el ejemplo mencionado, el almacenamiento debe ser en las bases de datos que manejen gran volumen de datos. En vista de que es un ejemplo sencillo no "técnico", se almacenarán los datos en la memoria de un computador.

[5] Revisar el Subcapítulo 2.3.2: Cluster
[6] Revisar el Subcapítulo 2.3.3: Cloud
[7] Revisar el Subcapítulo 2.3.4: Bases de datos relacionales
[8] Revisar el Subcapítulo 2.3.5: Bases de datos NoSQL

2.1.3. Análisis

El análisis es una de las áreas más importantes de Big Data, ya que se puede extraer información de valor que podría parecer oculta en el almacenamiento de datos. Pero la clave para obtener datos de valor es procesar la información de manera eficaz y en un tiempo razonable, de tal manera, que se puedan obtener resultados óptimos. No obstante, la mayoría de herramientas que existen trabajan únicamente con datos estructurados y otras suelen ser predefinidas y lentas al encontrarse con datos de gran volumen. Ante esto, los expertos recomiendan utilizar aplicaciones diseñadas específicamente para Big Data, para poder aprovechar al máximo esa capacidad ágil y proactiva para el análisis de datos.

Algunas de esas herramientas son, Apache Hadoop, “un framework de código abierto para el procesamiento, el almacenamiento y el análisis de grandes volúmenes de datos de diversas fuentes”. (Tablet Army, 2012)

Otra herramienta muy utilizada es ***R***, “un lenguaje de programación que facilita tanto el análisis de datos como el desarrollo de nuevo software de estadística”. (Tablet Army, 2012)

Esta área de Big data se encarga de extraer información relevante hacia el usuario. En este ejemplo se utiliza el programa R, que permite extraer información del archivo obtenido de OpenFlights. Para la demostración del siguiente ejemplo resulta relevante extraer información como latitud, y longitud de los aeropuertos en el mundo.

Para ello es necesario leer el archivo:

el siguiente comando permite visualizar los datos.

$$> \mathrm{x} = \mathrm{read.table}(\text{"airports.txt"}, \mathrm{header} = \mathrm{T}, \mathrm{sep} = \text{","})$$

$$> \mathrm{attach}(\mathrm{x})$$

$$> \mathrm{x}$$

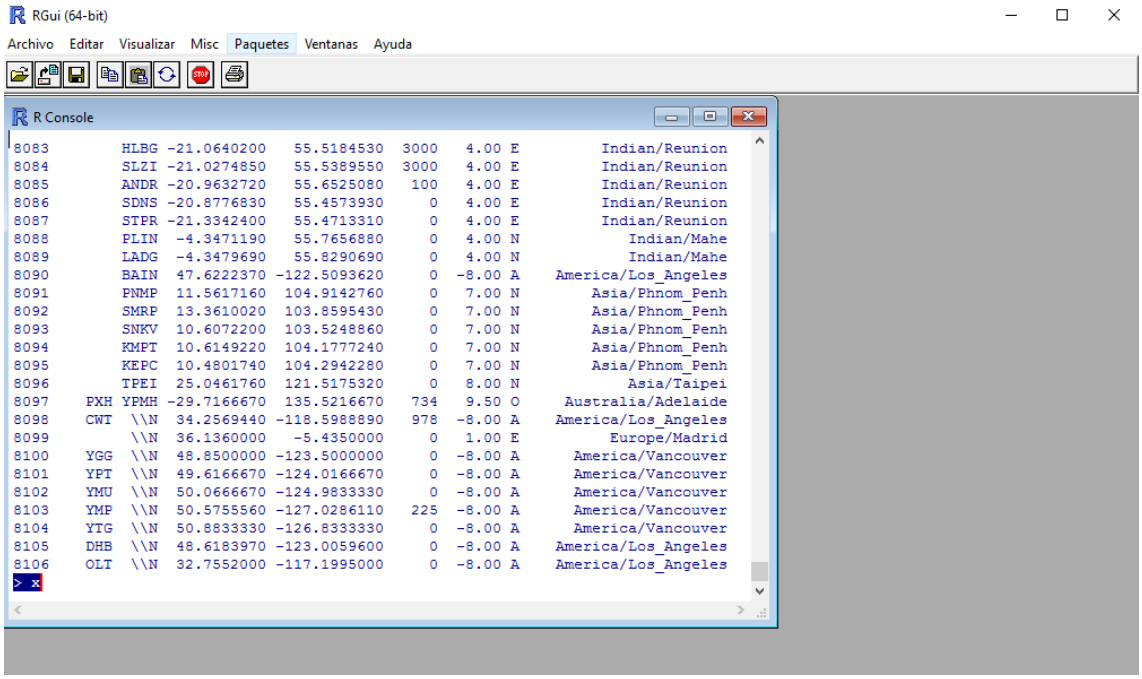


Figura 2.2: Visualización de datos en R.

Otro ejemplo consiste en utilizar un programa llamado ArGis que representa la información geográfica como una colección de capas y otros elementos en un mapa. Es así, que se ha recurrido a una imagen obtenida de google Earth localizada en Atacames-Ecuador.

El ejercicio consiste en identificar sectores de la imagen, o puntos relevantes donde se identifique la clasificación de los sectores a analizar, para ello se utilizan puntos con códigos que los representen, en este caso los códigos utilizados son los siguientes: 1 representa las vías, código 2 representa la arena de la playa, código 3 representa el mar, código 4 representa la piscina en la localidad, código 5 representa los árboles del sector. De esta manera se obtienen estos puntos:

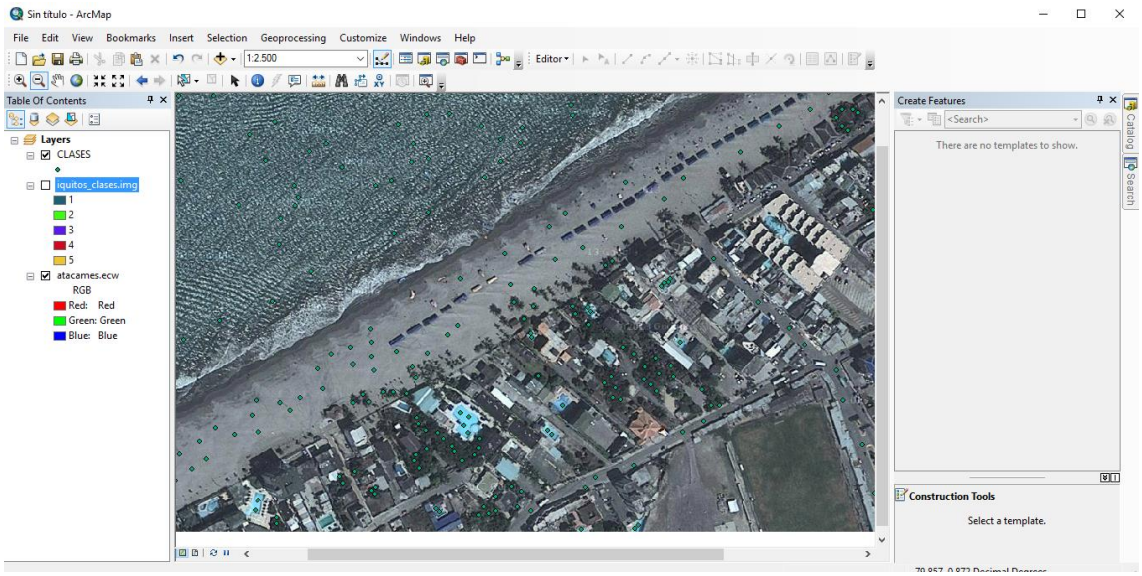


Figura 2.3: Visualización de Atacames en ArcGis.

Como se puede observar en la imagen, al lado izquierdo se muestran los códigos descritos anteriormente (1,2,3,4,5) mismos que están representados con distintos colores.

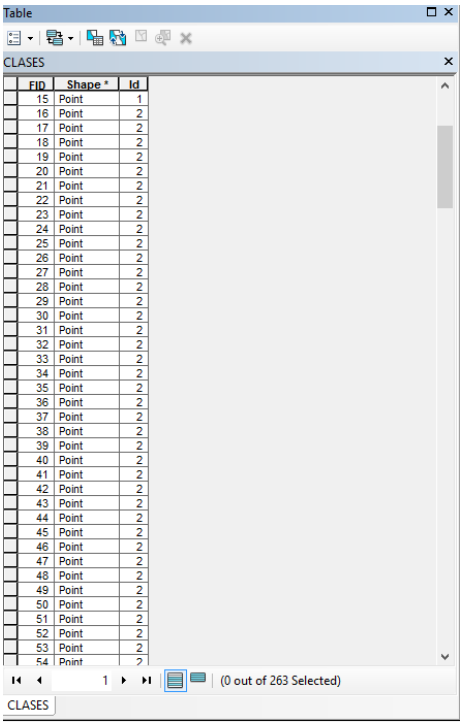


Figura 2.4: Visualización de los puntos recogidos.

2.1.4. Visualización

Los gráficos, mapas interactivos son la herramienta más utilizada para mostrar el análisis de los datos, permite difundir el análisis previo de manera precisa y consistente, para que posteriormente sea visualizada con las partes interesadas, por otro lado la visualización de datos ayuda a elaborar mejores cuadros de mando, y en general a comunicar el significado de los datos de la manera más adecuada para cada interlocutor.

Ahora es importante mencionar el ejemplo, para ello es necesario recordar los puntos recogidos en el área anterior. Los puntos permitirán identificar completamente las áreas clasificadas en la sección anterior. En esta sección se visualizarán todas las áreas verdes, todas las piscinas, todas las carreteras, etc. Como resultado se obtienen las áreas ya identificadas. Es así, que a continuación se muestra la imagen resultante que identifica ya todas las áreas.

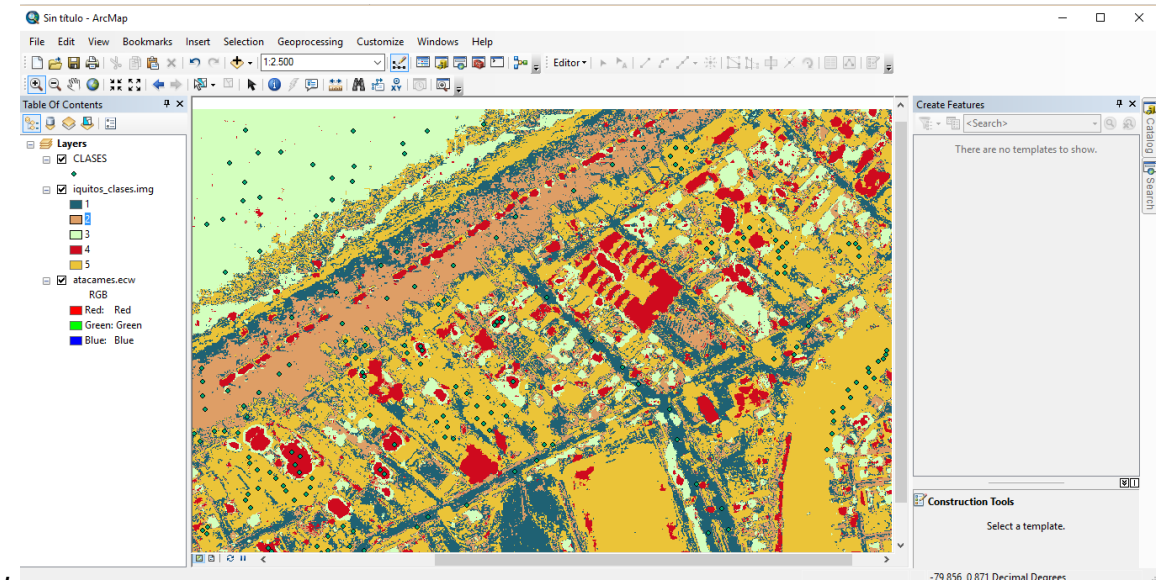


Figura 2.5: Visualización de las áreas identificadas.

2.2. Paradigmas de Big Data

Con la aparición de Big Data han surgido nuevos paradigmas de programación para facilitar el proceso y manejo de datos propiciando un acercamiento a una solución para Big Data. Los dos paradigmas que centran en el desarrollo de aplicaciones, así como en la gestión de los

grandes datos son MapReduce y Procesamiento Masivo en Paralelo (Massivelly Parallel Processing) o también conocido como MPP, ambos con aspectos en común pero bien diferenciados.

2.2.1. MapReduce

MapReduce es un modelo de programación que gestiona grandes volúmenes de datos, actualmente se ha implementado en varios sistemas, incluida la aplicación interna de Google. Igualmente, Hadoop puede usar una implementación de MapReduce para gestionar cálculos a gran escala de una manera que es tolerante a fallos de hardware. Para ello es necesario dos funciones, llamadas (Map) y (Reduce), mientras que el sistema se encarga de la ejecución en paralelo.

En resumen un MapReduce se ejecuta de esta manera:

1. Se presentan algún número de tareas, cada tarea se presenta en fichas distribuidas. Estas tareas de Mapa (Map) se convierten en pedazos de una secuencia que tienen clave/valor. Los pares de clave/valor se producen a partir de las entradas.
2. Los pares clave/valor de cada tarea de Mapa (Map) son recogidos de acuerdo a las claves y clasificadas. Las claves están distribuidas entre todos para las tareas de reducción (Reduce).
3. Las tareas reducidas trabajan sobre una clave a la vez, y combinan todos los valores asociados con aquella clave de algún modo. La forma de combinación se encuentra determinada por el usuario. (Leskovec, Rajaraman , & Ullman, 2014)

En la siguiente imagen encontraremos el procedimiento plasmado en forma gráfica.

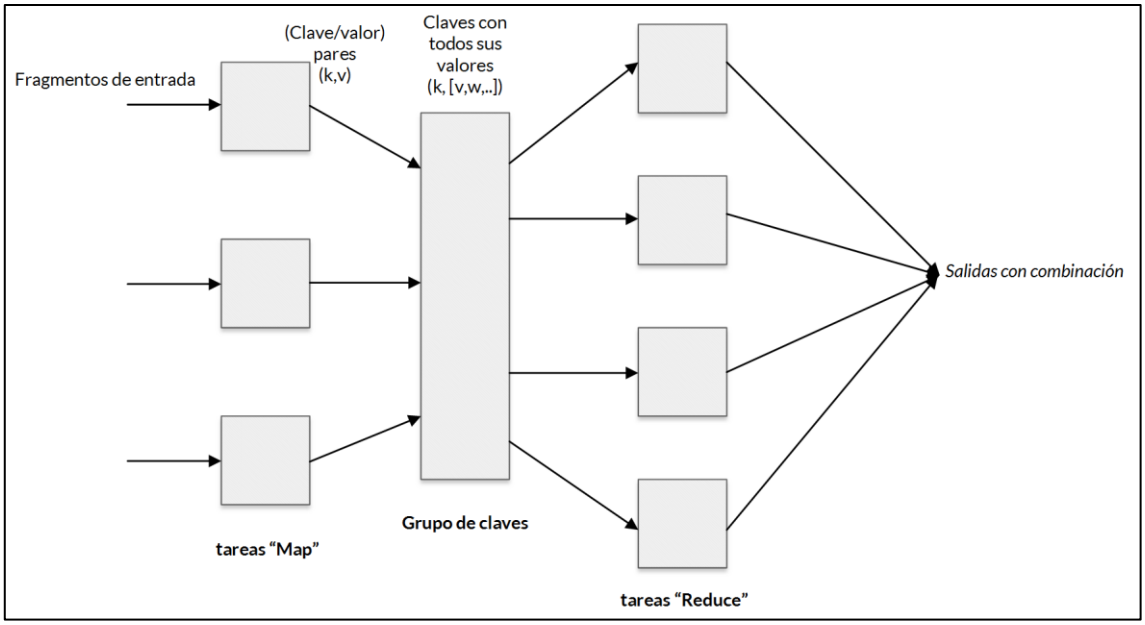


Figura 2.6: Esquema de un cálculo MapReduce. Fuente: (Leskovec, Rajaraman , & Ullman, 2014)

2.2.1.1. Tareas "Map"

Se considera que los archivos de entrada para una tarea "Map", pueden ser un conjunto de elementos de cualquier tipo:

Tabla 2.1: Tipos de datos en el paradigma Big data. Fuente: (Jiménez, Big data. Un nuevo paradigma, 2014)

Datos estructurados	Datos semiestructurados	Datos no estructurados
Fichas de clientes **Fecha de nacimiento** **Nombre** **Dirección** **Transacciones en un mes** **Puntos de compra**	**Correos electrónicos** Parte estructurada: destinatario, receptores, tema Parte no estructurada: cuerpo del mensaje	**Persona a persona** Comunicaciones en las redes sociales **Persona a máquina** Dispositivos médicos Comercio electrónico Ordenadores, móviles **Máquina a máquina** Sensores, dispositivos GPS Cámaras de seguridad

La función Map recibe como parámetros un par de (clave, valor) y devuelve una lista de pares. Esta función se encarga del mapeo y se aplica a cada elemento de la entrada de datos. Después se agrupan todos los pares con la misma clave de todas las listas, creando un grupo por cada una de las diferentes claves generadas. No hay requisito de que el tipo de datos para la entrada coincida con la salida y no es necesario que las claves de salida sean únicas.

$$(w_1, 1), (w_2, 1), \ldots\ldots\ldots,(w_3, 1)$$

2.2.1.2. Grupo de claves

Tan pronto como las tareas "Map" se han completado con éxito, los pares de clave/valor se agrupan por clave y los valores asociados a cada clave están formando una lista de valores. La agrupación se realiza por el sistema, independientemente "Map" y "Reduce" trabajan en las tareas. El controlador Master del proceso sabe cuándo aplicar "Reduce" a las tareas, pueden ser *r* tareas. El usuario normalmente indica al sistema MapReduce, que desea tener. A continuación, el controlador maestro escoge una **función hash**[9] que se aplica a las claves y produce una serie de números de *0* a *r-1*. Cada clave que se emite por una tarea de "Map" es ordenada y su par clave/valor se pone en uno de los *r* archivos locales. Cada expediente se destina a una de las tareas "Reduce".

Para llevar a cabo la agrupación de claves y la distribución de las tareas "Reduce", el controlador principal se fusiona con los archivos de cada tarea "Map" que están destinados a tareas particulares de "Reduce" y permiten la fusión de archivos. A ese proceso se le conoce como una secuencia de pares de claves que pertenecen a una lista de valores. Es decir, para cada clave *k*. La entrada para la función "Reduce" será *k*, las tareas formarán pares $(k, [v_1, v_2, \ldots., v_n])$, donde (k, v_1) , (k, v_2) ,….., (k, v_n) están en todos los pares clave/valor, y la clave *k* proviene de todas las tares "Map".

2.2.1.3. Tarea "Reduce"

[9] "Función Hash: Una función criptográfica hash- usualmente conocida como "hash"- es un algoritmo matemático que transforma cualquier bloque arbitrario de datos en una nueva serie de caracteres con una longitud fija. Independientemente de la longitud de los datos de entrada, el valor hash de salida tendrá siempre la misma longitud. De manera opcional, los usuarios pueden especificar su propia función hash u otro método de asignación de claves para reducir las tareas. Sin embargo, cualquiera que sea el algoritmo se utiliza, cada clave está asignada a una y sólo una tarea Reduce." (Brian Donohue, 2014)

El argumento de la función “Reduce” es un par formado por una clave y una lista de valores asociados. La salida de la función “Reduce” es una secuencia de cero o más pares de clave y valor. Estos pares de clave/valor pueden ser de un tipo diferente a los enviados en un principio por las tareas “Map” y “Reduce”, pero a menudo son del mismo tipo. Nos referiremos a la aplicación de la función “Reduce” a una única clave y una lista asociada de valores reductora. Una tarea “Reduce” recibe una o más claves y sus listas de valores asociados. Es decir, una tarea “Reduce” ejecuta uno o más reductores. Los resultados de todas las tareas “Reduce” se fusionan en un solo archivo. Los Reductores pueden ser divididos entre un número más pequeño de tareas “Reduce”, las claves se asocian a cada tarea “Reduce” con una serie de funciones hash.

El siguiente ejemplo muestra el proceso de MapReduce. Un caso de uso frecuente es aplicar un “Map” y “Reduce” de forma sucesiva, en primer lugar la preparación de un conjunto de datos a través de “Map”, y luego extraer alguna información a través de reducciones “Reduce”. El siguiente trabajo MapReduce cuenta las ocurrencias de cada palabra en algunos datos de entrada dados, como resultado final se obtendrán el número de palabras que se repiten en el documento. Para esto, el proceso sería el siguiente: como elemento de entrada se tiene un documento extenso que contiene distintas palabras, la función “Map” lee la entrada y produce pares de clave/valor. El paso siguiente es agrupar por claves, como está ilustrado en la figura 2.1. Finalmente, la función “Reduce” recoge todos los valores que pertenecen a la clave. La función “Map” y “Reduce” son definidos por las necesidades del programador, es así, que pueden existir varias formas de aplicar estas funciones.

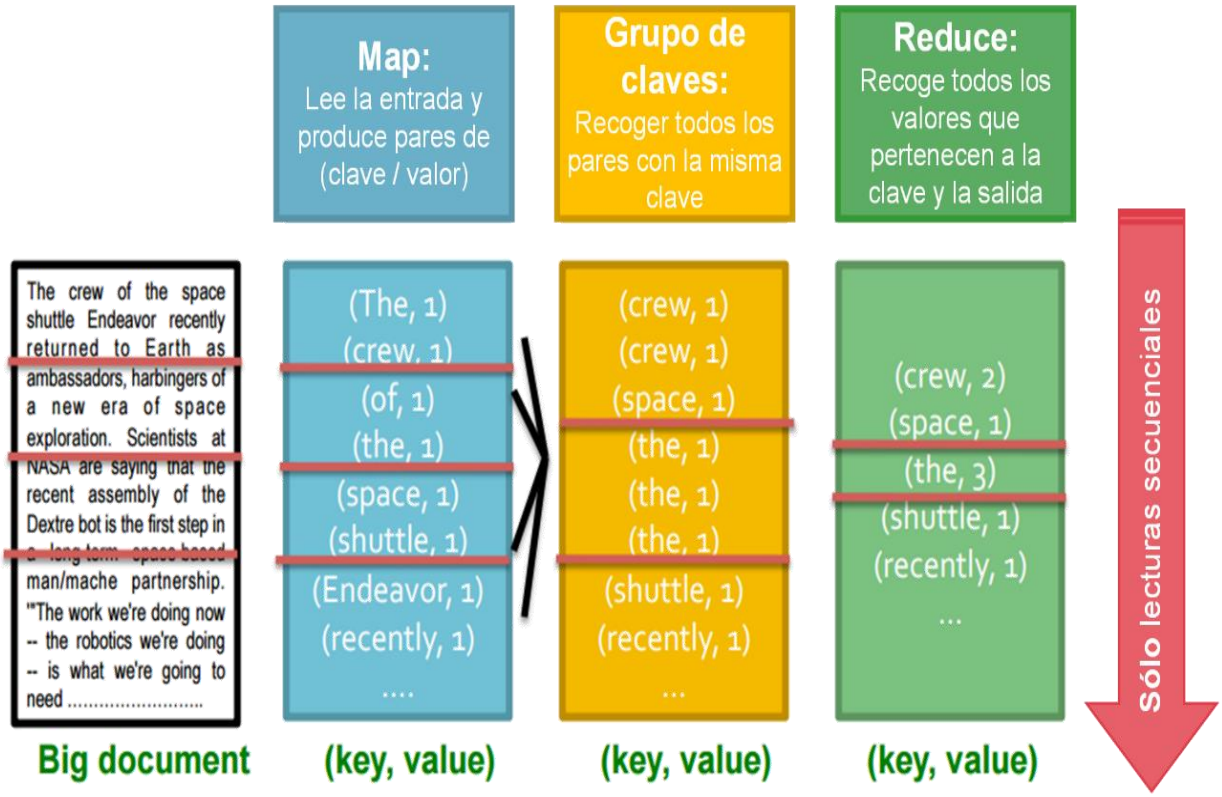


Figura 2.7: Esquemática de un cálculo MapReduce. Fuente: (Leskovec R. a.)

2.2.1.4. Ejemplo de un Algoritmo Usado Por MapReduce

Como un ejemplo, las principales operaciones en datos de Amazon implican responder a búsquedas para los productos, el registro de ventas, y así sucesivamente, procesos que implican relativamente poco cálculo y producen un cambio de la base de datos. Por otro lado, podría Amazon utilizar MapReduce para realizar ciertas consultas analíticas de grandes cantidades de datos, tales como la búsqueda de cada usuario. Aquellos usuarios cuyos patrones de compra eran muy similares. El propósito original para el cual la aplicación de Google de MapReduce se creó fue la de ejecutar muy grandes multiplicaciones de matriz-vector que sean necesarias en el cálculo del PageRank. Se expresará que matriz-vector-matriz y los cálculos de la matriz encajan muy bien en el estilo de computación de MapReduce. Otra clase importante de las operaciones que pueden utilizar MapReduce efectivamente son las operaciones del álgebra relacional.

2.2.1.4.1. Multiplicación por MapReduce

Supongamos que tenemos una matriz M de $n \times n$, cuyo elemento de la fila i y la columna j se denotará m_{ij}. Supongamos también que tenemos un vector v de longitud n, cuyo elemento j es de orden v_j . A continuación, el producto de matriz-vector es el vector x de longitud n, cuyo elemento x_i está dada por:

$$x_i = \sum_{j=1}^{n} m_{ij} v_j$$

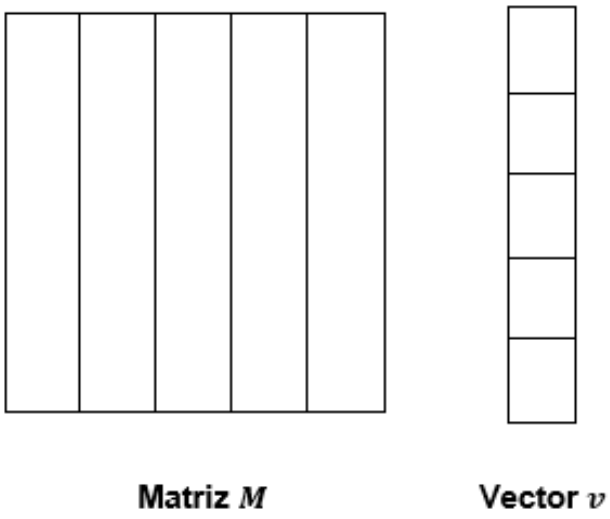


Figura 2.8: Visualización de Matriz M y Vector v

La matriz M y el vector v cada uno serán almacenados en un archivo DFS. Suponemos entonces, que las coordenadas de fila-columna de cada elemento de la matriz será visible, ya sea desde su posición en el archivo, o porque se almacena con coordenadas explícitas, tal como un triple (i, j, m_{ij}). También asumimos que la posición del elemento v_j en el vector v será visible en forma análoga.

Función "Map": La función "Map" se escribe para aplicar a un elemento de M. Sin embargo, si la lectura v no está ya en la memoria principal en el nodo de cómputo ejecutando una tarea "Map", entonces v es la primera lectura, en su totalidad, y posteriormente está disponible para todas las aplicaciones de la función "Map" realizado en esta tarea "Map". Cada tarea "Map" operará en un pedazo de la matriz M. De cada elemento de la matriz m_{ij} que produce el par

clave-valor $(i, j\, m_{ij} v_j)$. Entonces, todos los términos de la suma que conforman el componente x_i del producto matriz-vector tendrán la misma clave, i.

Función "Reduce": La función "Reduce" simplifica las sumas de todos los valores asociados con una clave dada i. El resultado será un par (i, x_i).

2.2.1.4.2. Multiplicación de Matrices

$$p_{ik} = \sum_j m_{ij} n_{jk}$$

Se requiere que el número de columnas de M sea igual al número de filas de N, por lo que la suma sobre j tiene sentido. Se puede pensar en una matriz como una relación con tres atributos: el número de fila, el número de columna, y el valor de esa fila y columna. Por lo tanto, se podría ver la matriz M como una relación $M\ (I, J, V)$, con tuplas (i, j, m_{ij}), y que se podría ver la matriz N como una relación $N\ (J, K, W)$, con tuplas (j, k, n_{jk}).

Sin embargo, es posible que i, j, k ya estén implícitas en la posición de un elemento de la matriz en el archivo que lo representa, en lugar de estar escritas de forma explícita con el elemento en sí. En ese caso, la función "Map" tendrá que ser diseñada para construir los componentes $I, J\ y\ K$ de tuplas de la posición de los datos.

La función "MAP": Esta función es sólo la identidad. Es decir, para cada elemento de entrada con la clave (i, k) y el valor de v, se produce exactamente este par clave-valor.

Función "Reduce": Para cada clave (i, k), se produe la suma de la lista de los valores asociados con esta clave. El resultado es un par $((i, k), v)$, donde $\boldsymbol{v}$ es el valor del elemento de la fila i y la columna k de la matriz $P = MN$.

Para poder entender con claridad cómo funciona la multiplicación de matrices con MapReduce se piensa que es necesario utilizar un ejemplo demostrativo que explique lo descrito anteriormente.

Ejemplo de Multiplicación de Matrices con MapReduce utilizando una matriz dispersa:

Objetivos:

- Demostrar la ejecución en paralelo.
- Demostrar la expresividad de MapReduce.

Ejecutar el producto entre C = A·B

Asumir que la mayoría de las entradas de la matriz son 0

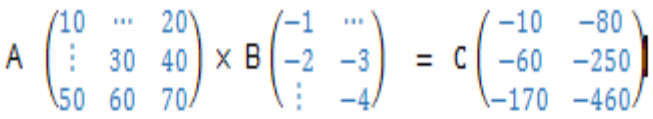


Figura 2.9: Matriz Dispersa.

Para poder realizar la multiplicación de matrices se debe tomar en cuenta que el número de columnas de la matriz A debe ser igual al número de filas de la matriz B. Entonces para la multiplicación se toman los valores de la primera fila de la matriz A y se los multiplica con la primera columna de la matriz B y así sucesivamente con todos los valores. Como se da en la siguiente demostración:

Primera Fila Matriz A: 10 0 20
Primera Columna Matriz B: -1 -2 0
Proceso: (10*-1 + 0*-2 + 20*0)
Resultado posición (1,1) de la Matriz C: -10

Segunda Fila Matriz A: 0 30 40
Primera Columna Matriz B: -1 -2 0
Proceso: (0*-1 + 30*-2 + 40*0)
Resultado posición (2,1) de la Matriz C: -60

Tercera Fila Matriz A: 50 60 70
Primera Columna Matriz B: -1 -2 0
Proceso: (50*-1 + 60*-2 + 70*0)
Resultado posición (3,1) de la Matriz C: -170

Primera Fila Matriz A: 10 0 20
Segunda Columna Matriz B: 0 -3 -4
Proceso: (10*0 + 0*-3 + 20*-4)
Resultado posición (1,2) de la Matriz C: -80
Segunda Fila Matriz A: 0 30 40
Segunda Columna Matriz B: 0 -3 -4
Proceso: (0*0 + 30*-3 + 40*-4)

Resultado posición (2,2) de la Matriz C: -250

Tercera Fila Matriz A: 50 60 70

Segunda Columna Matriz B: 0 -3 -4

Proceso: (50*0 + 60*-3 + 70*-4)

Resultado posición (2,2) de la Matriz C: -460

Resultado final Matriz C:

$$C \begin{pmatrix} -10 & -80 \\ -60 & -250 \\ -170 & -460 \end{pmatrix}$$

Figura 2.10: Resultado Matriz C.

Representar a la matriz como una lista de elementos no nulos (fila, columna, el valor, matriz ID)

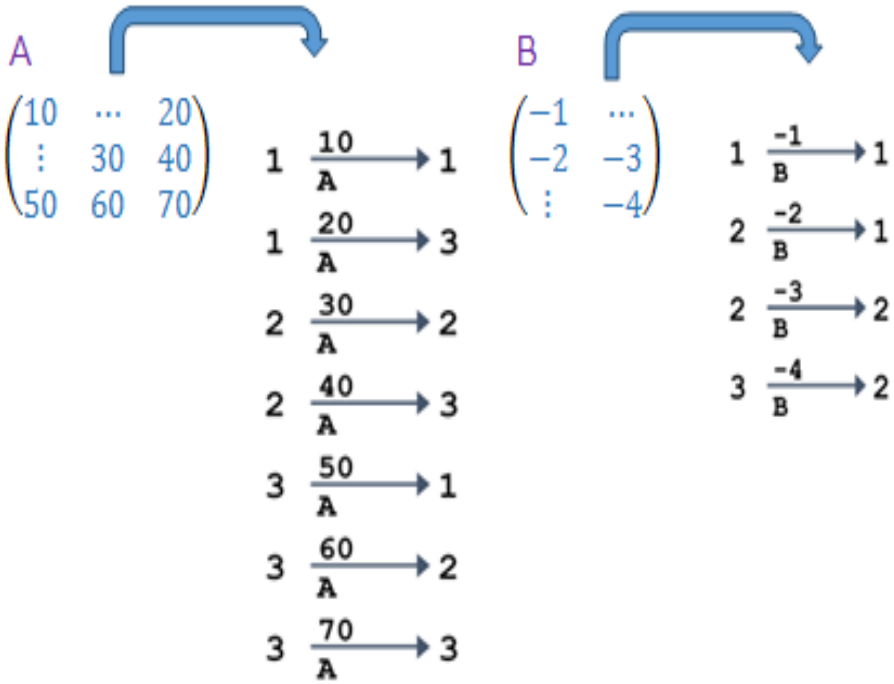


Figura 2.11: Representación de la Matriz como una lista de elementos no nulos.

Fase 1 Tarea "Map" de la Multiplicación de la Matriz

Se agrupan los valores $a_{i,k}$ y $b_{k,j}$ de acuerdo con la clave k

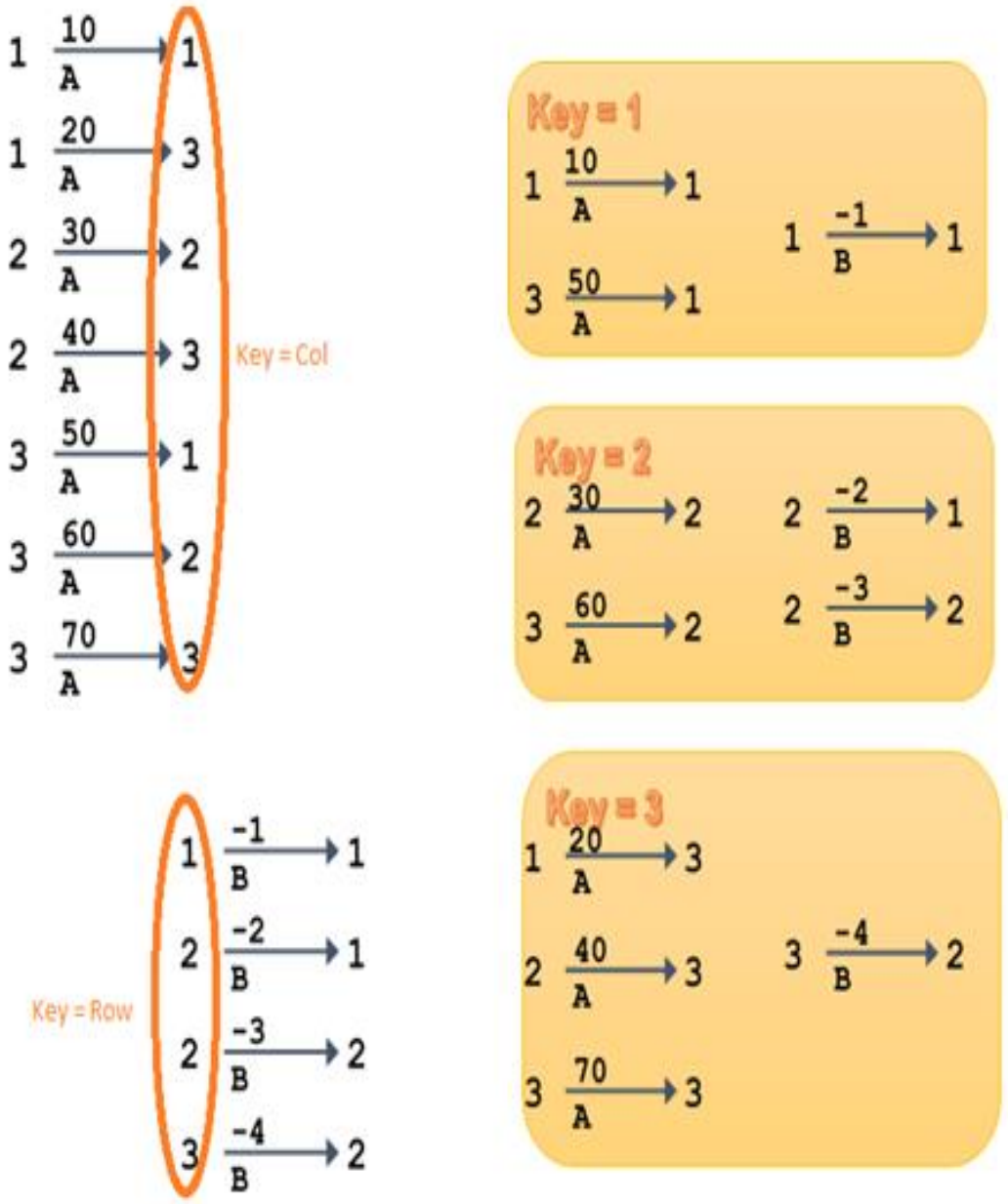


Figura 2.12: Tarea "Map" de la multiplicación de la matriz.

Fase 1 Tarea "Reduce" de la Multiplicación de la Matriz

Se generan todos los productos $a_{i,k} \cdot b_{k,j}$

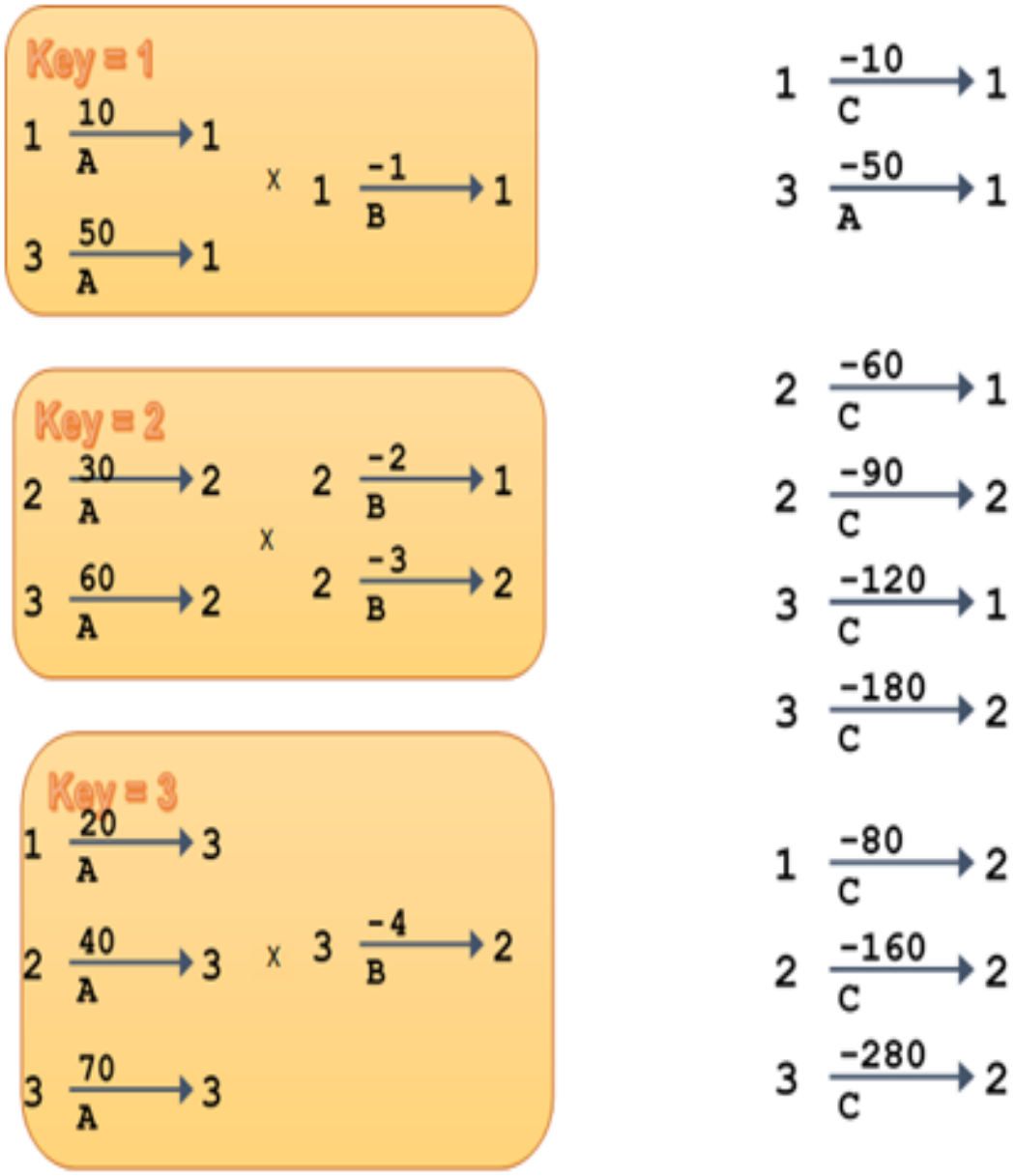


Figura 2.13: Tarea "Reduce" de la Multiplicación de la matriz.

Fase 2 Tarea "Map" de la Multiplicación de la Matriz

Agrupar los productos de $a_{i,k} \cdot b_{k,j}$ con el juego de valores de i y j.

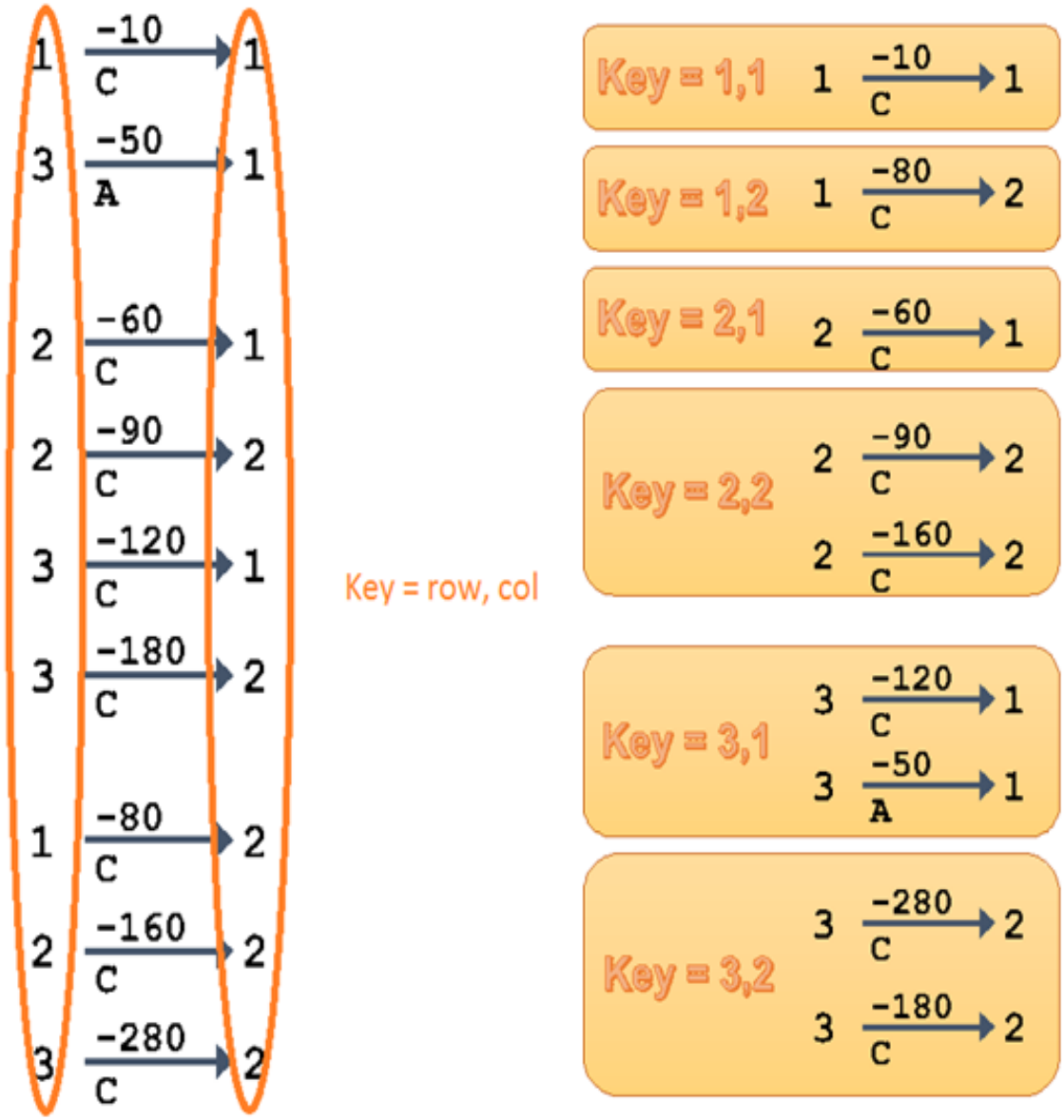


Figura 2.14: Tarea "Map" de la multiplicación de la matriz.

Fase 2 Tarea "Reduce" de la Multiplicación de la Matriz

Suma de los productos para obtener las entradas definitivas.

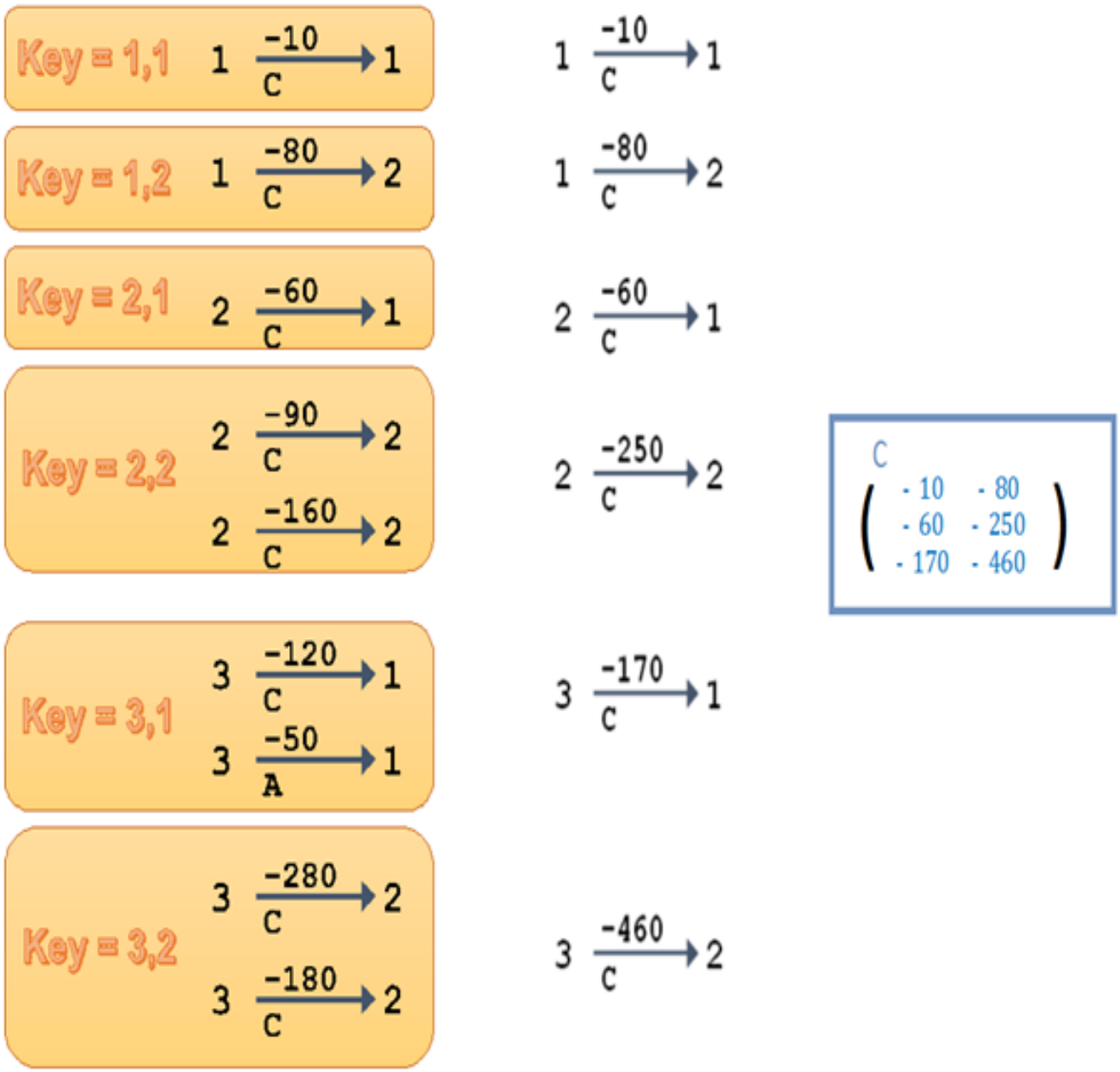


Figura 2.15: Resultado Final de la multiplicación de matrices.

Así se obtiene el resultado final que es la Matriz **C** del producto **A·B**

2.2.2. Procesamiento Masivo en Paralelo (Massively Parallel Processing – MPP)

Existe otro paradigma o método de cálculo para el procesamiento de consultas distribuidas y es el llamado Procesamiento Masivo en Paralelo o MPP, éste es muy similar a MapReduce pero cuando se trata de procesar y analizar grandes volúmenes de datos, la base de datos MPP resulta mejor debido a su mayor rapidez en comparación con Hadoop utilizada por MapReduce, además que realiza análisis más complejos que combinan varios conjuntos de datos diferentes. “En MPP, como en MapReduce, el procesamiento de los datos se distribuye a través de un banco de nodos de computación, estos nodos separados procesan sus datos en paralelo y los conjuntos de salida de nivel de nodo se ensamblan entre sí para producir un conjunto de resultados finales”. (Brust, 2012)

MPP es el procesamiento coordinado de un programa de múltiples procesadores trabajando en diferentes partes del programa. Cada procesador tiene su propio sistema operativo y su respectiva memoria. MPP acelera el rendimiento de enormes bases de datos que tienen que gestionar cantidades masivas de datos.

Las bases de datos MPP utilizan procesadores de múltiples núcleos, múltiples procesadores, servidores y dispositivos de almacenamiento, equipados para el procesamiento en paralelo. Esa combinación permite leer muchas piezas de datos a través de muchas unidades de procesamiento, mejorando al mismo tiempo la velocidad.

2.3. Conceptos Generales del Capítulo

2.3.1. Sistema de ficheros distribuidos.

Un sistema de ficheros distribuidos se puede definir como, un conjunto de computadores interconectados que comparten un estado, ofreciendo una visión de sistema único. (Alberto Lafuente, 2007)

2.3.2. Clúster

Es un conjunto de ordenadores conectados entre sí y con un software específico que les permite trabajar simultáneamente proporcionando una mayor capacidad de cómputo. (Heredero, 2004)

2.3.3. Cloud Computing

Consiste en la posibilidad de ofrecer servicios a través de Internet. La computación en la nube, es una nueva tecnología que busca tener toda la información, ya sea personal o publica, en Internet y sin depender de límites de almacenamiento. El Cloud Computing explica las nuevas posibilidades de forma de negocio actual, ofreciendo servicios a través de Internet, conocidos como e-business (negocios por Internet). (Debitoor, 2010)

2.3.4. Bases de Datos

Se puede decir que una base de datos es un banco de información que contiene datos importantes relacionados con diversas temáticas y que se encuentran clasificados de distinta manera, pero al mismo tiempo comparten mutuamente algún tipo de relación que busca ordenarlos y clasificarlos. Es así, que existen diferentes tipos de bases de datos que manejan y trabajan con los datos de diferente manera.

2.3.4.1. Bases de Datos Relacionales

Una base de datos relacional es una colección de elementos de datos organizados en un conjunto de tablas formalmente descritas desde la que se puede acceder a los datos o volver a montarlos de muchas maneras diferentes sin tener que reorganizar las tablas de la base. La base de datos relacional fue inventada por E.F. Codd en IBM en 1970. (Margaret Rouse, 2015)

2.3.4.2. Bases de Datos OLTP Y OLAP

“Las bases de datos tradicionales almacenan transacciones que se refieren al trasciego de información operacional de una organización, es decir, operaciones que se llevan a cabo diariamente. Estos sistemas se denominan **OLTP (On-Line Transaction Processing, procesamiento transaccional en línea).** Por ejemplo, un cajero automático de un banco es demostración de una aplicación OLTP ya que se deben guardar cada una de las transacciones realizadas.

Sin embargo, los sistemas OLTP no están preparados para el análisis de los datos registrados. Un analista que quiera acceder a los datos históricos de una organización para poder tomar decisiones necesita de sistemas con otro tipo de requisitos diferentes a los de OLTP. Estos sistemas se denominan **OLAP (On-Line Analytical Processing,**

procesamiento analítico en línea) y hacen uso de bases de datos multidimensionales para incrementar la capacidad de análisis de los usuarios. Por ejemplo, un analista bancario podría necesitar estudiar las transacciones en los cajeros automáticos para determinar las comisiones a cobrar minimizando el coste a los usuarios, pero sin que el banco tenga pérdidas. Este análisis no se puede llevar a cabo directamente sobre el sistema OLTP porque resultaría costoso, por lo que se debe diseñar una base de datos multidimensional que permita el análisis de los datos mediante herramienta OLAP." (Trujillo, Diseño y explotación de almacenes de datos: conceptos básicos de modelado multidimensional, 2013)

2.3.4.3. Modelo Multidimensional

Un modelo multidimensional se representa en forma de cubo o hipercubo (cubo sobre cubo) o en su versión más sencilla, como tablas multidimensionales (tipo hoja de cálculo). Un ejemplo de cubo se puede ver en la fig. 2.8; y un ejemplo de tabla multidimensional en la fig. 3.2, donde tenemos un hecho de ventas de productos a ser analizado por almacén, producto y fecha en que se realizan las ventas. (Trujillo, Diseño y explotación de almacenes de datos: conceptos básicos de modelado multidimensional, 2013)

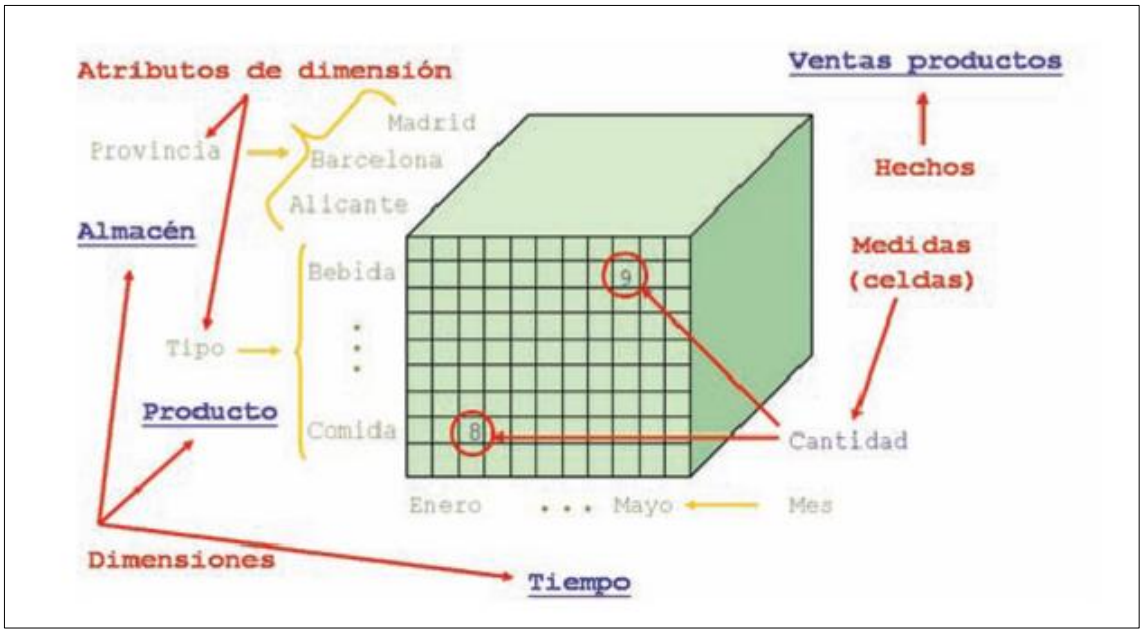


Figura 2.16: Ejemplo de cubo de datos. Fuente: (Trujillo, Diseño y explotación de almacenes de datos: conceptos básicos de modelado multidimensional., 2013)

Ventas			Producto.Grupo = "Supermercado"			
			Comida		Bebida	
			Cong	Fresco	Refresco	Alcohol
Almacén. comunidad = "Comunidad Valenciana"	Alicante	Albatera	100	200	300	400
		Elche	500	600	700	800
	Valencia	Burjasot	900	1000	1100	1200
		Cullera	1300	1400	1500	1600

Figura 2.17: Ejemplo de tabla multidimensional. Fuente: (Trujillo, Diseño y explotación de almacenes de datos: conceptos básicos de modelado multidimensional., 2013)

2.3.4.4. Bases de Datos NoSQL

EL pradigma NoSQL. NoSQL no es un sustituto a las bases de datos relacionales, es solo un movimiento que busca otras opciones para escenarios específicos, "No uses sólo SQL". Históricamente, el término fue primero usado en los 90's para nombrar una base de datos relacional open source. Sin embargo, como denominador del conjunto de bases de datos alternativas al modelo relacional, fue primero usado en 2009 por Eric Evans para nombrar una serie de conferencias sobre este tipo de bases de datos. Aunque el término más correcto sería NoREL (Not Only Relational), como varios han señalado, el término NoSQL ya tiene gran aceptación. (Camacho, 2010)

CAPITULO III: Herramientas y tecnologías de Big Data

El objetivo de este capítulo es conocer las plataformas que permiten establecer ambientes estables de Big Data, así como sus respectivas herramientas.

3.1. Plataformas de Big Data

Alrededor de los últimos 15 a 20 años muchas empresas y organizaciones han estado trabajando con una arquitectura de datos que manejaba bases de datos relacionales de tipo OLTP (On-Line Transaction Processing, procesamiento transaccional en línea). Este tipo de arquitectura funcionaba perfectamente cuando se trabajaba con gigabaytes y terabytes de datos estructurados, pero lamentablemente para que los usuarios pudieran obtener reportes y resultados de diferentes consultas tenían que esperar semanas o incluso meses, lo cual resultaba perjudicial para los usuarios.

Por el contrario, Google, Yahoo! y Facebook no pudieron acoplarse y trabajar con esta tecnología, es así, que se desarrolló una nueva generación de recursos para la administración y el análisis de datos, siendo algunos de ellos proyectos de código abierto, para que los desarrolladores de estas grandes compañías innovadoras pudieran actualizar y mejorar las capacidades de estas herramientas de administración y análisis mucho antes que cualquier otro proveedor. De este modo, estas empresas se vieron en la necesidad de adquirir plataformas que les ayuden en el manejo y análisis de varios tipos de datos y grandes volúmenes.

En la siguiente sección se hablará de las principales plataformas que hacen Big Data.

3.1.1. Apache Hadoop

“Es un entorno de desarrollo de código abierto que soporta de manera nativa aplicaciones distribuidas, en paralelo y que hacen un uso de datos intensivo. Para muchos, Hadoop se ha convertido en un sinónimo de Big Data. Soporta la ejecución de aplicaciones en grandes clusters de hardware dedicado empleando una arquitectura de escalabilidad horizontal. Hadoop implementa un paradigma de programación llamado MapReduce, en el que la aplicación se divide en muchos pequeños fragmentos de tareas, donde cada cual se puede ejecutar o volver a ejecutar en cualquier nodo del cluster (el sistema de archivos distribuidos de Hadoop, o HDFS), que almacena datos en los nodos del ordenador y que proporciona un

ancho de banda agregado en todo el cluster. Tanto MapReduce como HDFS están diseñados de modo que el entorno de trabajo gestiona automáticamente los fallos de nodo. Hace posible que las aplicaciones funciones con miles de ordenadores que trabajan de modo independiente y con petabytes de datos. Actualmente se considera que la "plataforma" completa de Apache Hadoop consiste en el **kernel**[10] de Hadoop, MapReduce, HDFS y varios relaciones, en los que se incluyen Apache Hive, y Apache HBase." (Schmarzo, Apache Hadoop, 2013)

3.1.2. Apache Spark

"Apache Spark es un potente motor de procesamiento de código abierto construido en torno a la velocidad, facilidad de uso y análisis sofisticados. Originalmente fue desarrollado en la Universidad de Berkeley en 2009." (Matei Zaharia, 2014)

"Spark es una plataforma de computación de código abierto para análisis y procesos avanzados, que tiene muchas ventajas sobre Hadoop. Desde el principio, Spark fue diseñado para soportar en memoria algoritmos iterativos que se pudiesen desarrollar sin escribir un conjunto de resultados cada vez que se procesaba un dato. Esta habilidad para mantener todo en memoria es una técnica de computación de alto rendimiento aplicado al análisis avanzado, la cual permite que Spark tenga unas velocidades de procesamiento que sean 100 veces más rápidas que las conseguidas utilizando MapReduce. Spark tiene un framework integrado para implementar análisis avanzados que incluye la librería MLlib, el motor gráfico GraphX, Spark Streaming, y la herramienta de consulta Shark. Esta plataforma asegura a los usuarios la consistencia en los resultados a través de distintos tipos de análisis. " (O'Ryan, 2014)

[10] "Kernel ó núcleo, es un software que constituye una parte fundamental del sistema operativo. Es el principal responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora él es el encargado de gestionar recursos, a través de servicios de llamada al sistema, también se encarga de decidir qué programa podrá hacer uso de un dispositivo de hardware y durante cuánto tiempo, lo que se conoce como multiplexado. Acceder al hardware directamente puede ser realmente complejo, por lo que los núcleos suelen implementar una serie de abstracciones del hardware. Esto permite esconder la complejidad, y proporciona una interfaz limpia y uniforme al hardware subyacente, lo que facilita su uso al programador." (EcuRed, 2016)

3.1.3. Oracle Big Data Appliance

El Oracle Big Data Appliance es un sistema de ingeniería que combina hardware y software, puede ser utilizado para capturar y analizar grades datos de amplia variedad.

El producto incluye una distribución de código abierto de Apache Hadoop, cuenta con la base de datos Oracle NoSQL, Oracle Data integrador con adaptador Solicitud de Hadoop. Es preinstalado y pre configurado con **Cloudera CDH**[11]. Además de eso, el precio del hardware (US $ 525.000 para un sistema de rack completo, incluye el costo de Cloudera CDH y sus opciones de Cloudera Manager.

Es así que por $ 525.000 se obtiene el siguiente:

- Big Data Appliance Hardware (viene con la solicitud de servicio automático a fallos de los componentes)
- Cloudera CDH y Cloudera Administrador
- Todas las opciones de Cloudera, así como Accumulo y Spark (CDH) 5.0
- Oracle Linux y Oracle JDK
- Distribución de Oracle R
- Base de datos de Oracle NoSQL Community Edition
- Oracle Big Data Appliance Enterprise Manager plug-in

Además, el costo de soporte es de $63.000 por año, a continuación se muestra la lista de precio para un el servicio de Premier Support por 3 años, incluyendo el precio del equipo, el precio de la instalación y el servicio de soporte.

Tabla 3. 1: Costos de infraestructura y servicio Premier Support. Fuente: (Jean-Pierre Dijcks-Oracle, 2014)

	Año 1	año 2	año 3	Costo total
Costo BDA	$ 525,000			
Costo Anual de Apoyo	$ 63.000	$ 63.000	$ 63.000	
Instalar en el lugar (aproximadamente)	$ 14.000			
Total	$ 602,000	$ 63.000	$ 63.000	$ 728,150

[11] **CDH** es la distribución de la plataforma de código abierto 100% de Cloudera, incluyendo Apache Hadoop y construido específicamente para satisfacer las demandas de la empresa. CDH ofrece todo lo necesario para el uso empresarial sacarlo de la caja. Mediante la integración de Hadoop con más de una docena de otros proyectos de código abierto críticos, Cloudera ha creado un sistema funcionalmente avanzado que lo ayuda a hacer de extremo a extremo de los flujos de trabajo de grandes datos. (Cloudera, 2015)

Por ese precio usted recibirá un estante pre integrado con las siguientes especificaciones.

Tabla 3. 2: Elementos de una implementación de grandes volúmenes de datos Mediana Empresa de costo / beneficio. Fuente: (Nik Rouda, Senior Analyst and Adam DeMattia, Research Analyst, 2015)

Item	Valor	Métricas
Hardware/ Network		
Nodos	18	Servidores – Cada 2 x 18 núcleos con procesadores Intel Xeon
Núcleos	36	Por nodo
Memoria	128	GB/servidor
Racks	1	El estante puede abarcar hasta 18 nodos
Almacenamiento de nodos	96	TB/servidor para clúster primarios , almacenamiento interno
Administración de almacenamiento de información	50	Terabytes; asume una cuarta parte de los datos totales en el máximo movimiento en un determinado momento.
Switches	3	Infiniband. Mejora el rendimiento 3 veces más de 10GBe (Gigabit Ethernet). Además, cuanta un interruptor de administración y cableado variado.
Soporte de hardware	15%	Del costo total del hardware, la tercera parte de soporte.
Software		
Costo de licencia Hadoop	18 @ $7,200/nodo	Típicamente nuevas licencias durante la temprana adopción de datos grande, permiten dos licencias más para copia de seguridad.

Figura 3.1: Oracle Big Data Appliance Fuente: (Oracle, 2016)

3.2. Introducción a las Tecnologías de Big Data

Las tecnologías de Big Data actualmente, tienen el potencial de reforzar notablemente el almacenamiento y tratamiento millones de datos con las siguientes ventajas para la investigación y aplicación en diferentes campos como en Business Intelligence o en la estrategia de marketing. Es así, que las organizaciones tienen ahora la oportunidad de ampliar sus recursos de almacenamiento de datos si aprovechan las siguientes posibilidades:

Almacenamiento, acceso y análisis de enormes volúmenes de datos transaccionales estructurados tales como; ventas, pedidos, envió, transacciones, registros de call centers, transacciones de tarjetas de crédito.

Integración de datos semiestructurados, por ejemplo registros de sensores, GPS y datos telemétricos y datos sin estructurar como campos de texto, comentarios de consumidores, documentos y registros de mantenimiento, que aportan nuevas dimensiones, atributos dimensionales y nuevas métricas de informes.

Feeds de datos en tiempo real, acompañado de entornos analíticos en tiempo real para capturar, analizar, identificar y actuar sobre anomalías en datos conforme van llegando a las organizaciones.

Analítica predictiva que pueden ponderar; prever; detectar; dar predicciones y proveer recomendaciones. Por ejemplo alertas, informes y paneles de control.

A continuación, se muestran brevemente las principales tecnologías de Big Data de las herramientas descritas anteriormente para ser comparadas en un análisis posterior.

3.2.1. Tecnologías Relacionadas con Apache Hadoop

3.2.1.1. Apache Hive

"Apache Hive es una infraestructura de almacenamiento de datos basada en Hadoop que permite hacer resúmenes, consultas y análisis de datos. Aunque fue Facebook quien la desarrolló inicialmente, en la actualidad la utilizan y perfeccionan otras compañías, como Netflix. Apache Hive soporta análisis de grandes conjuntos de datos almacenados en sistemas de ficheros compatibles con Hadoop. Proporciona un lenguaje de tipo SQL, llamado HiveQL, y sigue manteniendo un soporte completo para Map Reduce. Para acelerar las consultas, Hive proporciona índices en los que se incluyen índices de mapas de bits.

3.2.1.2. Apache Hbase

HBase es un modelo de base de datos no relacional, distribuido y de código abierto escrito en Java. Fue desarrollado como parte del proyecto Apche Hadoop de la Apache Software Foundation y se ejecuta sobre HDFS. HBase proporciona un medio para almacenar grandes cantidades de datos dispersos tolerante a fallos. Las tablas de HBase pueden servir como entradas y salidas para las tareas MapReduce ejecutadas en Hadoop, y se puede acceder a ellas a través de la API de Java.

3.2.1.3. Pig

Pig es un entorno de trabajo y un lenguaje de programación de alto nivel que trabaja con tareas en paralelo pensado para crear programas para MapReduce. Pig abstrae el lenguaje de programación de MapReduce en construcciones de más alto nivel, de un modo similar a lo que ocurre con SQL y los sistemas de gestión de bases de datos relacionales. Pig se puede ampliar utilizando funciones definidas por el usuario, que el desarrollador puede escribir en

Java, Python, JavaScript o Ruby y luego llamar directamente o desde el lenguaje." (Schmarzo, Big Data: Understanding How Data Powers Big Business (Big Data, El poder de los datos), 2013)

3.2.2. Tecnologías Relacionadas Con Apache Spark

Existen una serie de herramientas adicionales que forman parte del ecosistema de Spark y le proporcionan características adicionales en la analítica de Big Data.

3.2.2.1. Spark SQL

Permite la consulta de datos estructurados utilizando lenguaje SQL o una API, que se puede usar con Java, Scala, Python o R.

3.2.2.2. Spark Streaming

Mientras MapReduce solo procesa datos en lotes, Spark tiene la posibilidad de gestionar grandes datos en tiempo real. Esto facilita que los datos se analicen según van entrando, sin tiempo de latencia y a través de un proceso de gestión en continuo movimiento.

3.2.2.3. Spark MLlib (Machine Learning)

Esta herramienta contiene algoritmos que dotan a Apache Spark de muchas utilidades, como la regresión logística y máquinas de vectores de soporte (SVM); modelos de árbol de regresión bayesiana; técnicas de mínimos cuadrados; modelos de mezclas gausianas; análisis de conglomerados de K medias; asignación latente de Dirichlet (LDA); descomposición en valores singulares (SVD); análisis de componentes principales (ACP); regresión lineal; regresión isotónica.

3.2.2.4. Spark Graphx

Es un framework de procesamiento gráfico. Proporciona una API para la elaboración de grafos con los datos.

3.2.3. Tecnologías Relacionadas con Oracle Big Data Appliance

3.2.3.1. Oracle Big Data SQL

"Oracle Big Data SQL integra datos a través de Hadoop, NoSQL, y Oracle Database y trabaja sobre Oracle Big Data Appliance para simplificar la búsqueda de datos almacenados. Con esto, se puede consultar y analizar datos a través de toda su gama de sistemas de gestión de datos." (Bécares, 2014)

3.2.3.2. Oracle NoSQL DataBase

"Oracle NoSQL Database proporciona un modelo de transacción poderosa y flexible que simplifica enormemente el proceso de desarrollo de una aplicación basada en NoSQL. Es una plataforma diseñada para ayudar a los usuarios a gestionar grandes archivos de datos no estructurados. La compañía ha incluido NoSQL como un componente de su sistema Big Data Appliance." Los datos pueden ser modelado como tablas de estilo de base de datos relacional, documentos JSON o pares de clave y valor. (Oracle, 2015)

3.2.3.3. Oracle Data Integrator (ODI)

"Oracle Data Integrator es una plataforma de integración completa que cubre los requisitos de integración de datos. Maneja alto volumen, provee lotes de alto desempeño a procesos dirigidos a eventos, a servicios de integración basados en una arquitectura orientada a servicios y con la capacidad de procesar eventos en tiempo real." (Angel Rios, 2009)

3.3. Tabla comparativa

Tabla 3.3: Tabla comparativa Apache Spark, Apache Hadoop, Oracle Big Data Appliance.

	Apache Spark	Apache Hadoop	Oracle Big Data Appliance
REQUISITOS			
Núcleos de CPU	8-16 núcleos por máquina	4 núcleos por máquina	36 / Nodo
Memoria	8 GB	24 GB	128 GB/Servidor
Almacenamiento	4-8 discos por nodo	4–6 discos 2TB por nodo	12 discos 4 TB por nodo
Red	10 GBe o más	1 GB Ethernet	10 GBe o más
CARACTERÍSTICAS GENERALES			
Costo	Open Source, pero existen costos asociados con el personal y hardaware requerido	Open Source, pero existen costos asociados con el personal y hardaware requerido	El precio del hardware es US $ 525.000 para un sistema de rack completo. A este precio se debe sumar el servicio y Cloudera

Seguridad	Se considera escasa en vista de que es un nuevo producto	Cuenta con Service Level Authorization, que asegura que los usuarios tengas los permisos correspondientes, se integra con proyectos de seguridad como Knox Gateway and Sentry	Ofrece seguridad empresarial integrada con autenticación Kerberos preconfigurada, autorización basada en LDAP y auditoría centralizada robusta conOracle Audit Vault y Database Firewall
Actuación	Funciona mejor cuando todos los datos caben en la memoria, especialmente en grupos dedicados	Está diseñado para los datos que no caben en la memoria y puede funcionar bien junto a otros servicios	El producto incluye una distribución de código abierto de Apache Hadoop e incluye funcionalidades propias de Oracle
Velocidad	Ejecuta 100 veces más rápido que Hadoop en memoria y 10 veces más rápido si el acceso es de Disco	Hadoop tiende a demorar más en su ejecución tanto desde memoria como Disco	Puede realizar la carga de datos en paralelo y de alta velocidad desde Hadoop a Oracle Database

CAPITULO IV: Instalación de un Ambiente de Big Data y Casos Prácticos.

El objetivo de este capítulo es elaborar una serie de procedimientos que detallen paso a paso el proceso de instalación de Hadoop, además se pretende explicar el funcionamiento de cada uno de los servicios con los que éste cuenta, esto proporcionará un fácil aprendizaje en lo que concierne a Hadoop.

4.1. Instalación de un Ambiente de Big Data

4.1.1. Selección de las herramientas

Para el desarrollo de este capítulo la principal herramienta que se utilizará para la creación de un ambiente de Big Data es Hadoop, que como ya se detalló en el Capítulo 3, Hadoop es una herramienta de software libre que permite el procesamiento distribuido de grandes volúmenes de datos mediante un clúster. Además, se escogió Hadoop en lugar de Oracle Big Data Apliance o Spark, debido a que Hadoop es mucho más intuitivo por lo tanto, resulta más fácil el manejo de esta herramienta.

Más adelante se detallarán los pasos de instalación de Hadoop, pero antes es necesario explicar qué modelo de arquitectura de clúster de Hadoop se utilizará para realizar la mencionada instalación.

Existen tres modelos de arquitecturas de un clúster de Hadoop:

- **Modo No Distribuido:**

El modo no distribuido también es conocido como modo de un solo nodo (single node), el cual se ejecuta como un solo proceso de JAVA y es más utilizado para depuración.

- **Modo Pseudo-distribuido:**

El modo pseudo-distribuido es aquel en el cual un único nodo es configurado para trabajar como una simulación de una arquitectura distribuida, es ideal para desarrollo y probar aplicaciones.

- **Modo Completamente Distribuido:**

El modo completamente distribuido es aquel en el cual un clúster se configura como una arquitectura distribuida con todos los servicios maestro-esclavos funcionando y es apropiado para un entorno de producción.

Entonces, la arquitectura del clúster escogida para la siguiente instalación de Hadoop es la de un clúster en modo Completamente Distribuido, en la cual se utilizará un computador que funcionará como Nodo Master, y tres computadores adicionales que funcionarán como Nodos Esclavo respectivamente. Así, el modelo de la arquitectura del clúster que se implementará queda organizado de la siguiente manera:

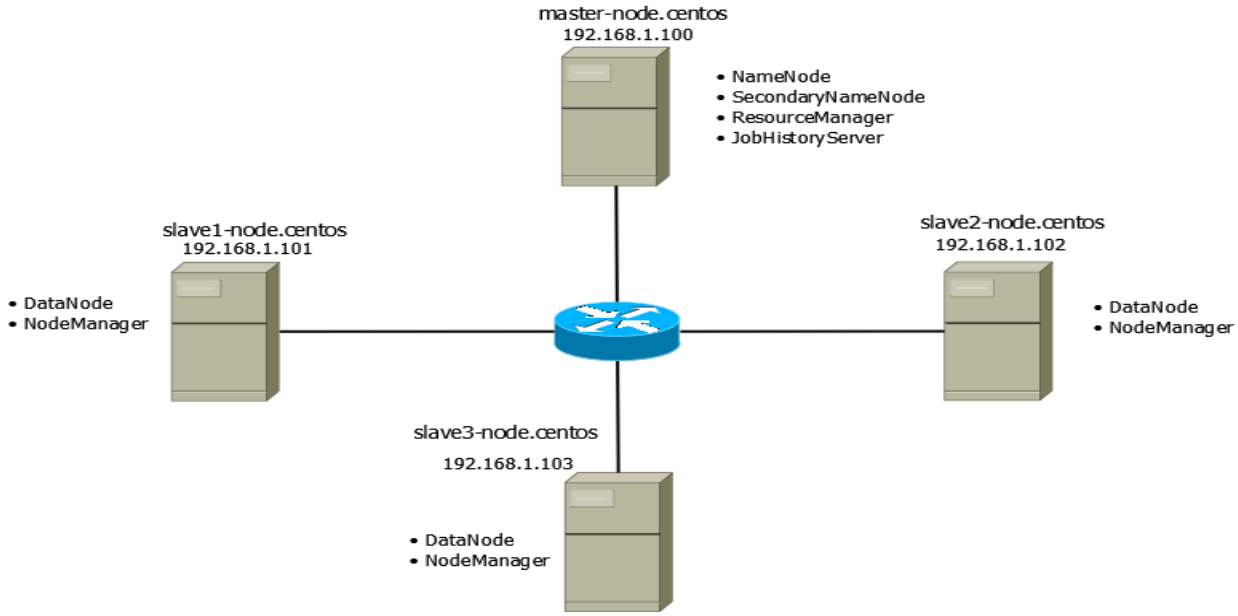


Figura 4.1: Topología de un clúster de Hadoop en Modo Completamente Distribuido (multi-nodo).

4.1.2. Requisitos de Instalación de Hadoop

- **Plataformas Compatibles**

✓ GNU / Linux es útil como plataforma de desarrollo y producción. Existen pruebas que Hadoop ha sido demostrado en los clústeres de GNU / Linux con 2000 nodos. (The Apache Software Foundation, 2016)

✓ Windows, también es una plataforma compatible aunque no es muy utilizada para la instalación de esta herramienta.

Para la siguiente instalación se utilizará la distribución de Linux: ***Centos 7***.

- **Software Necesario**

✓ JDK de JAVA.

4.1.3. Pasos de Instalación de Hadoop

En esta guía se explica claramente el funcionamiento de cada uno de los servicios y demonios que serán ejecutados en el clúster, es así, que existen dos formas de iniciar dichos servicios. Con la configuración del fichero *.bashrc* se omiten ciertos pasos los cuales son: paso 12, 13, 14, 15, 16 y 17.

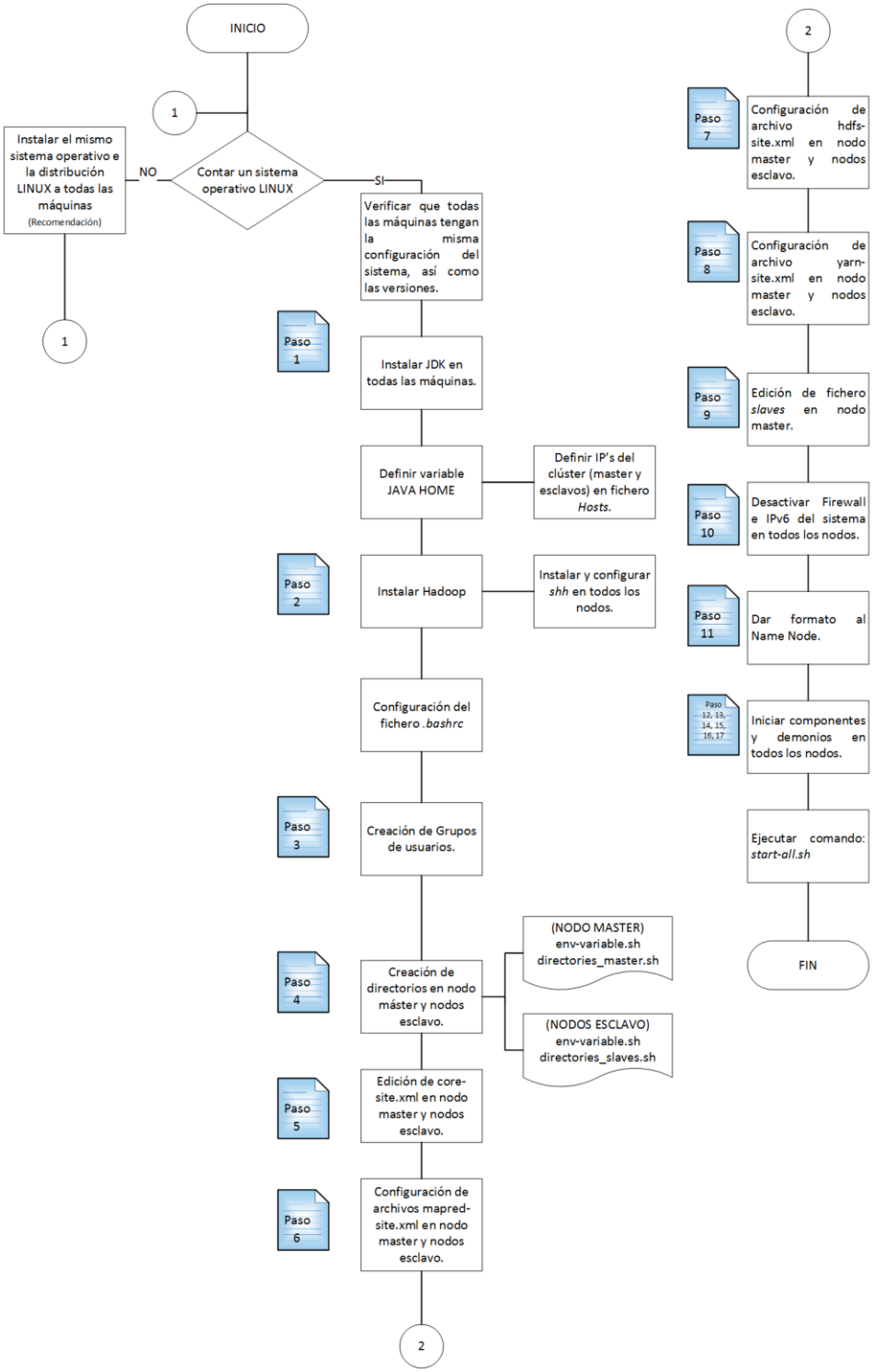


Figura 4.2: Diagrama de Flujo del proceso de Instalación de Hadoop.

PASO 1:

Instalación del JDK

Antes de instalar Hadoop, es necesario tener instalado el JDK de Java, para ello es necesario descargar el JDK de la página de Oracle (http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html?ssSourceSiteId=otnes). Es importante tener instalado el JDK en todas las máquinas, tanto para el nodo master como para los nodos esclavo.

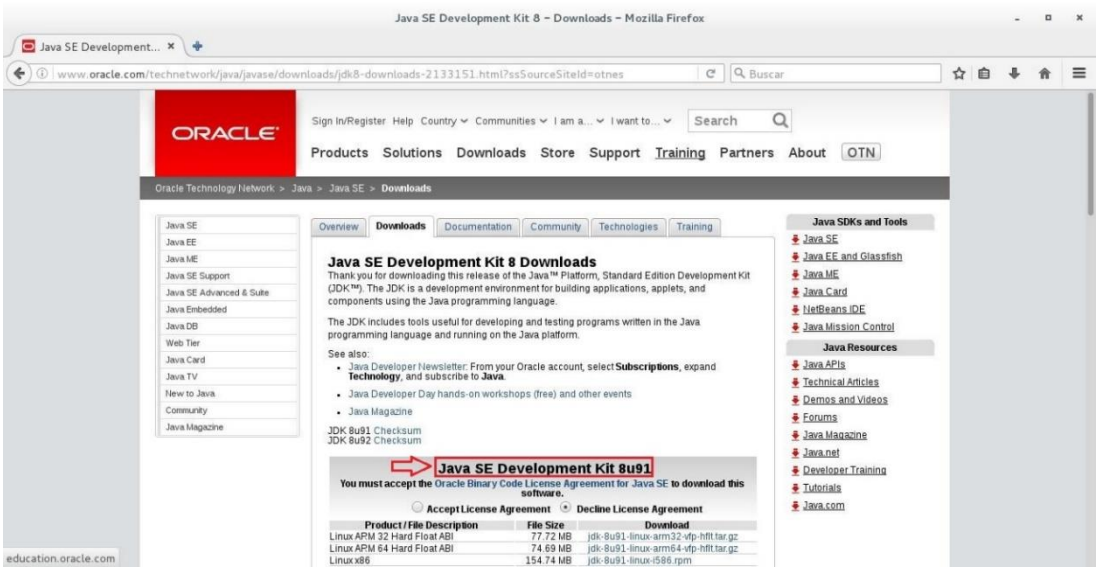


Figura 4.3: Jdk, Página web Oracle.

- Iniciar el Terminal, es necesario entrar con privilegios de *root* y estar ubicados en el directorio *root*.

Digitar los siguientes comandos:

- Para acceder al sistema con privilegios de root:

```
$su
```

- Para moverse al directorio root:

```
#cd
```

- Para confirmar el directorio en el cual se está trabajando:

```
#pwd
```

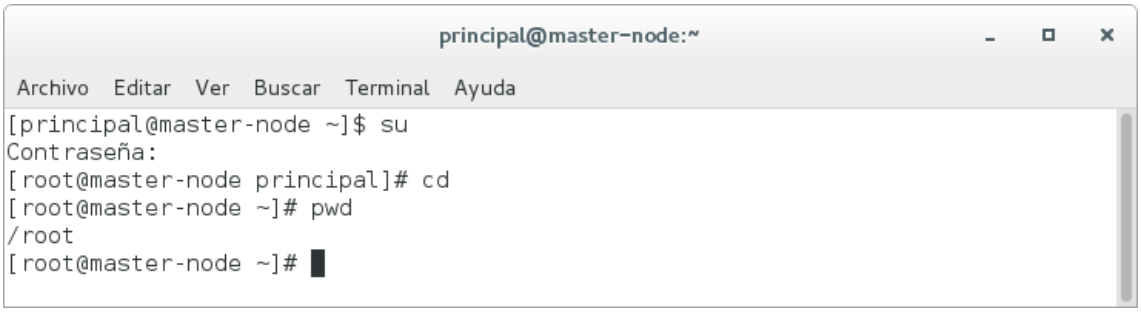


Figura 4.4: Acceso como root.

- Una vez descargado el JDK de Java, es necesario copiar el paquete descargado del JDK en el directorio *root*, debido a que todas las descargas realizadas se ubican en el directorio *Descargas.*

Digitar los siguientes comandos:

- Para copiar el paquete del JDK ubicado en el directorio *Descargas* al directorio *root*:

```
#cp /home/principal/Descargas/jdk-8u91-linux-x64.rpm /root/
```

- Para listar el contenido del directorio:

```
#ls
```

```
[root@master-node ~]# cp /home/principal/Descargas/jdk-8u91-linux-x64.rpm /root/
[root@master-node ~]# ls
anaconda-ks.cfg  initial-setup-ks.cfg  jdk-8u91-linux-x64.rpm
[root@master-node ~]#
```

Figura 4.5: Copiar jdk a la ubicación root.

Se puede observar que el paquete JDK se ha copiado correctamente en el directorio *root*, es por ello que aparece de color rojo.

- Ahora es necesario instalar el JDK con el siguiente comando, como ya se mencionó anteriormente este paso es importante realizarlo en todos los nodos:

```
#rpm -ivh jdk-8u91-linux-x64.rpm
```

```
[root@master-node ~]# rpm -ivh jdk-8u91-linux-x64.rpm
Preparando...                          ################################# [100%]
Actualizando / instalando...
   1:jdk1.8.0_91-2000:1.8.0_91-fcs     ###                               ( 13%)
```

Figura 4.6: Instalación del jdk.

- Esperar hasta que el proceso finalice.

```
[root@master-node ~]# rpm -ivh jdk-8u91-linux-x64.rpm
Preparando...                          ################################# [100%]
Actualizando / instalando...
   1:jdk1.8.0_91-2000:1.8.0_91-fcs     ################################# [100%]
Unpacking JAR files...
        tools.jar...
        plugin.jar...
        javaws.jar...
        deploy.jar...
        rt.jar...
        jsse.jar...
        charsets.jar...
        localedata.jar...
        jfxrt.jar...
[root@master-node ~]#
```

Figura 4.7: Instalación del jdk (proceso).

- Editar y definir la variable JAVA_HOME, para esto se escribe el siguiente comando, y se presiona Enter:

```
#nano /etc/profile
```

```
[root@master-node ~]# nano /etc/profile
```

Figura 4.8: Definir variable JAVA_HOME.

Al dar Enter se presenta en pantalla lo siguiente:

```
principal@master-node:~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
  GNU nano 2.3.1                Fichero: /etc/profile

# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what you
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.

pathmunge () {
    case ":${PATH}:" in
        *:"$1":*)
            ;;
        *)
            if [ "$2" = "after" ] ; then
                PATH=$PATH:$1
            else
                PATH=$1:$PATH

^G Ver ayuda   ^O Guardar     ^R Leer Fich   ^Y Pág Ant     ^K CortarTxt   ^C Pos actual
^X Salir       ^J Justificar  ^W Buscar      ^V Pág Sig     ^U PegarTxt    ^T Ortografía
```

Figura 4.9: Archivo para definir variable JAVA_HOME (1).

- Ir al final del texto, tal como se muestra el cursor en la siguiente imagen:

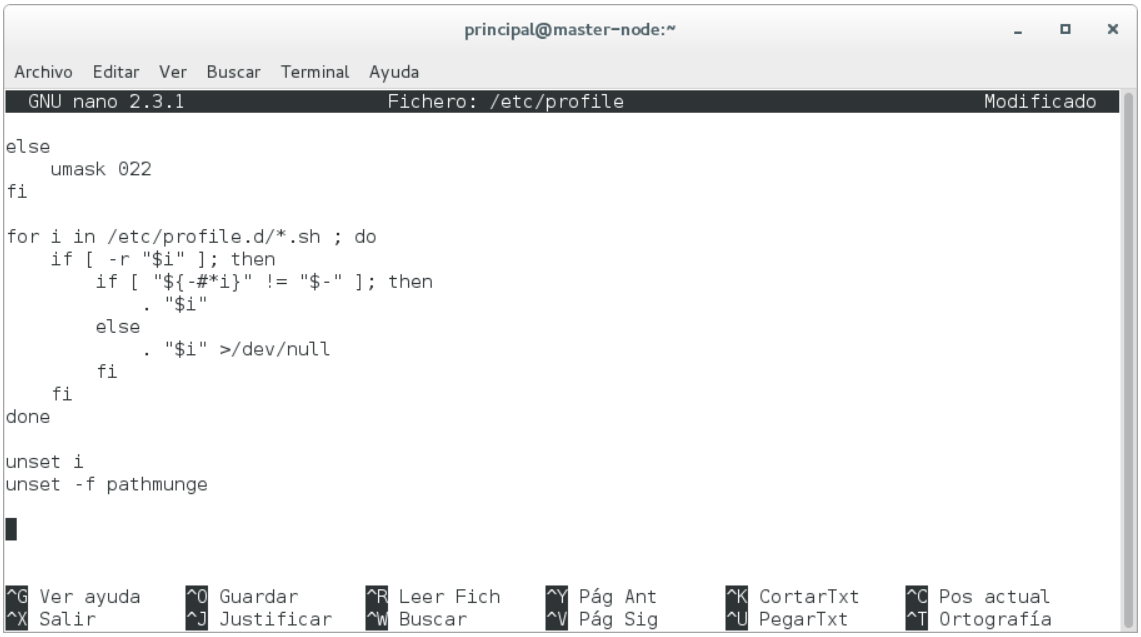


Figura 4.10: Archivo para definir variable JAVA_HOME (2).

- Para poder definir la variable JAVA_HOME, se digitan los siguientes comandos:

```
export JAVA_HOME=/usr/java/jdk1.8.0_91/
export PATH=$JAVA_HOME/bin:$PATH
```

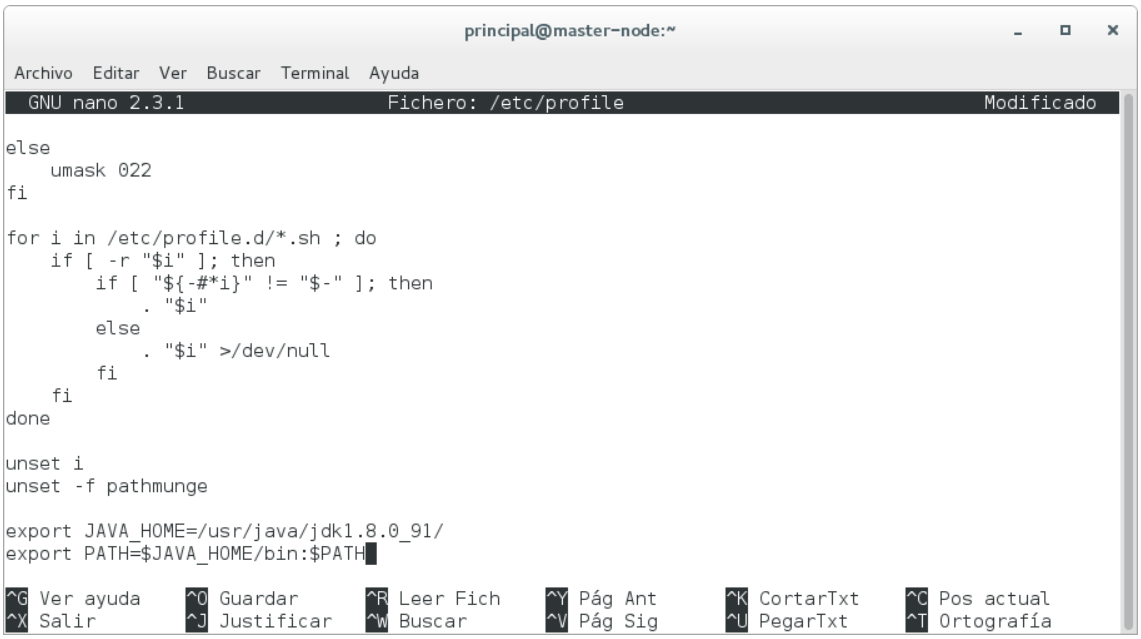


Figura 4.11: Comandos necesarios para definir variable JAVA_HOME.

Presionar Ctrl + X (para salir)

- Verificar si el PATH es un directorio, para ello se utiliza lo siguiente:

```
#. /etc/profile
#$JAVA_HOME
```

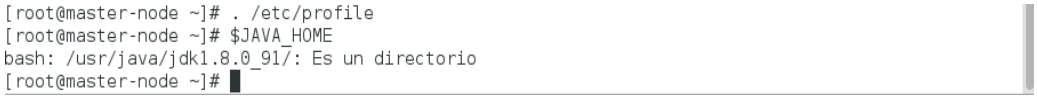


Figura 4.12: Comprobación Java es un directorio.

- Una vez instalo JAVA, se procede a editar tanto en el Nodo Master como en los Nodos Esclavo el archivo */hosts* para guardar la correspondencia entre dominios y las direcciones IP.

Digitar lo siguiente:

```
#nano /etc/hosts
```

Figura 4.13: Abrir archivo Hosts

Se abre el siguiente fichero:

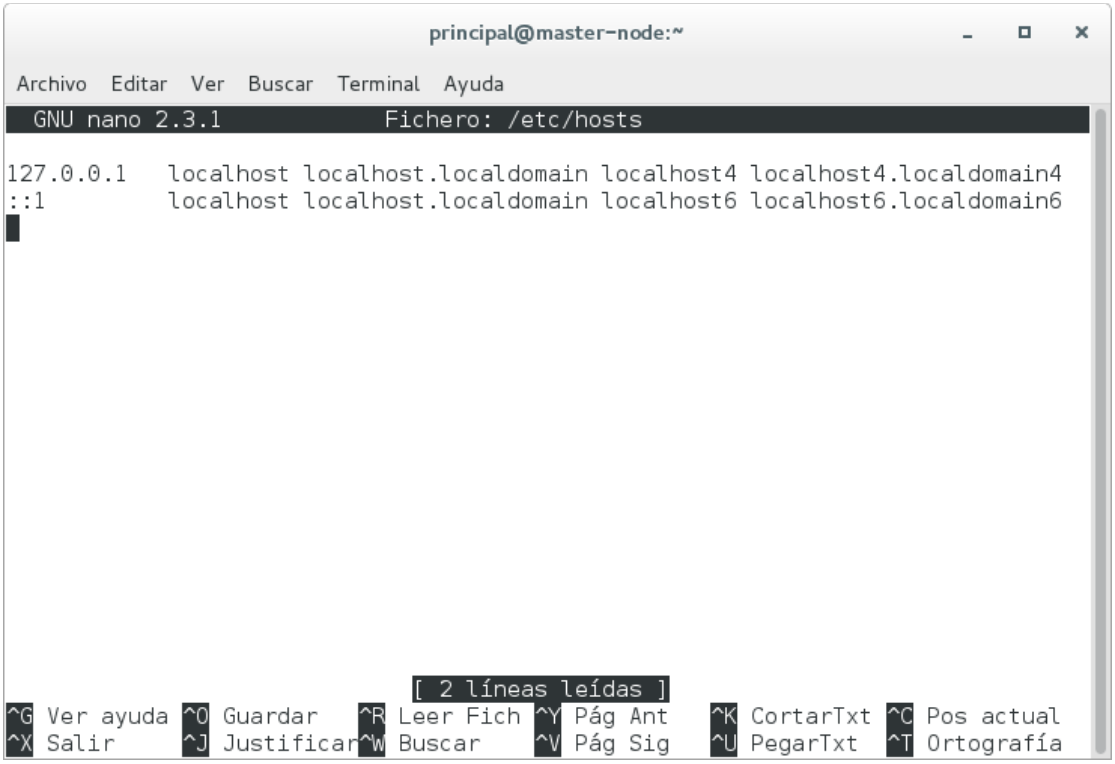


Figura 4.14: Archivo Hosts.

- Se agrega la IP y los dominios de cada uno de los nodos del Cluster:

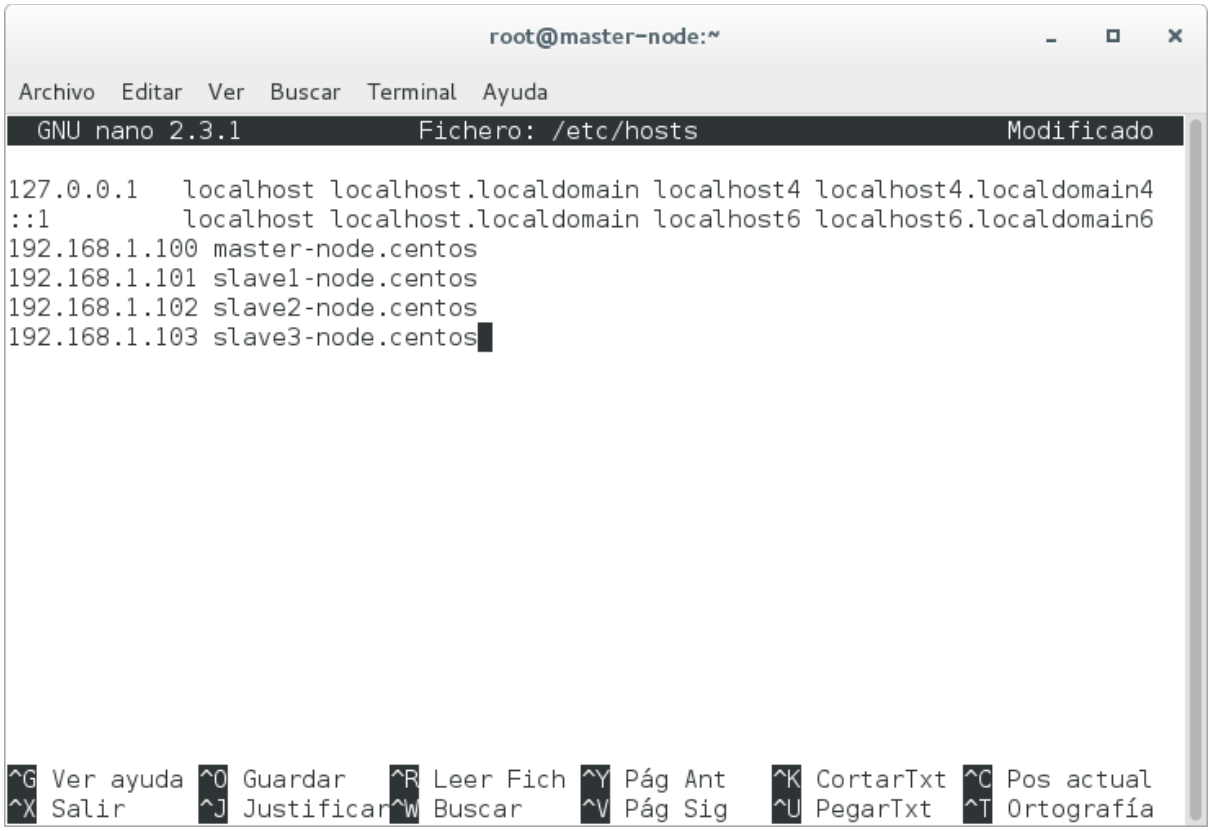


Figura 4.15: IP's y Dominios de los nodos del clúster.

Presionar Ctrl + X (para salir)

- En seguida, se realiza un ping para comprobar que los nodos se están comunicando entre ellos.

```
[root@master-node ~]# ping slave1-node.centos
PING slave1-node.centos (192.168.1.101) 56(84) bytes of data.
64 bytes from slave1-node.centos (192.168.1.101): icmp_seq=1 ttl=64 time=0.590 ms
64 bytes from slave1-node.centos (192.168.1.101): icmp_seq=2 ttl=64 time=0.511 ms
64 bytes from slave1-node.centos (192.168.1.101): icmp_seq=3 ttl=64 time=0.515 ms
^C
--- slave1-node.centos ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.511/0.538/0.590/0.045 ms
[root@master-node ~]# ping slave2-node.centos
PING slave2-node.centos (192.168.1.102) 56(84) bytes of data.
64 bytes from slave2-node.centos (192.168.1.102): icmp_seq=1 ttl=64 time=0.563 ms
64 bytes from slave2-node.centos (192.168.1.102): icmp_seq=2 ttl=64 time=0.531 ms
^C
--- slave2-node.centos ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.531/0.547/0.563/0.016 ms
[root@master-node ~]# ping slave3-node.centos
PING slave3-node.centos (192.168.1.103) 56(84) bytes of data.
64 bytes from slave3-node.centos (192.168.1.103): icmp_seq=1 ttl=64 time=0.527 ms
64 bytes from slave3-node.centos (192.168.1.103): icmp_seq=2 ttl=64 time=0.564 ms
64 bytes from slave3-node.centos (192.168.1.103): icmp_seq=3 ttl=64 time=0.506 ms
^C
--- slave3-node.centos ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.506/0.532/0.564/0.030 ms
[root@master-node ~]#
```

Figura 4.16: Ping de comprobación de conexión de los nodos.

PASO 2:

Instalación De Hadoop

- Descargar Hadoop del siguiente link: http://www-eu.apache.org/dist/hadoop/common/hadoop-2.7.2/. Se debe tomar en cuenta que hadoop debe ser descargado e instalado tanto en el Nodo Master como en los Nodos Esclavo.

Figura 4.17: Página de descarga de Hadoop.

Se escoge la versión ***hadoop-2.7.2.tar.gz***

- Tal como se hizo con el paquete del JDK, se copia el archivo *.tar* de Hadoop que está ubicado en el directorio *Descargas* al directorio *root.*

Se digitan los siguientes comandos:

- Para copiar el archivo *.tar* de Hadoop al directorio *root*:

```
#cp /home/principal/Descargas/hadoop-2.7.2.tar.gz /root/
```

- Para listar el contenido del directorio:

```
#ls
```

```
[root@master-node ~]# cp /home/principal/Descargas/hadoop-2.7.2.tar.gz /root/
[root@master-node ~]# ls
anaconda-ks.cfg  hadoop-2.7.2.tar.gz  initial-setup-ks.cfg  jdk-8u91-linux-x64.rpm
[root@master-node ~]#
```

Figura 4.18: Copiar Hadoop al directorio root.

- A continuación, se descomprime el archivo *.tar* que contiene Hadoop y se lo copia a la ubicación donde se lo quiera instalar.

Se digitan los siguientes comandos:

- Para descomprimir el archivo *.tar* de Hadoop:

```
#tar xzf hadoop-2.7.2.tar.gz
```

- Para copiar el archivo hadoop-2.7.2 a la ubicación en la cual se desea realizar la instalación:

```
#cp -R hadoop-2.7.2 /opt/hadoop
```

```
[root@master-node ~]# tar xzf hadoop-2.7.2.tar.gz
[root@master-node ~]# cp -R hadoop-2.7.2 /opt/hadoop
```

Figura 4.19: Descomprensión de Hadoop y Copia de Hadoop al directorio Hadoop.

Configuración De Hadoop y SSH

La configuración de Hadoop incluye setear las variables de entorno de Hadoop en el archivo *.bashrc.* El objetivo de esta configuración es poder iniciar los servicios y demonios de Hadoop, tanto del nodo master como de los nodos esclavos desde la máquina correspondiente al nodo master sin necesidad de ejecutar estas tareas por individual en cada máquina. Motivo por el cual, también es necesario realizar la configuración *ssh* en todas las máquinas, para poder

acceder de manera remota a cada una de ellas. El manejo de *ssh* puede ser tanto con contraseña como sin ella, para esta práctica se utilizará *ssh* con una configuración sin contraseña, lo cual permitirá acceder desde la máquina del nodo master a las máquinas correspondientes a los nodos eslavo para poder iniciar y ejecutar más fácilmente los demonios y servicios de Hadoop.

Para poder entender este funcionamiento es necesario conocer tanto lo que es el fichero *.bashrc* y cómo trabaja *ssh (Secure Shell – intérprete de órdenes seguro)* en un clúster.

Fichero *.bashrc*: El fichero .bashrc es un archivo que se lo puede encontrar en la carpeta personal de cada usuario en la siguiente dirección dependiendo de la configuración de cada máquina: */home/nombre_de_usuario/.bashrc.* Este archivo, contiene las configuraciones de inicio de programas, herramientas y es en donde se pueden setear las variables de servicios como Hadoop y Java para que sus aplicaciones puedan ser accedidas y ejecutadas de manera más rápida.

SSH: Secure Shell o intéprete de órdenes seguro también conocido por sus siglas SSH, es un protocolo que facilita la comunicación entre dos o más sistemas, es decir, permite la conexión entre diferentes máquinas a un host de manera remota.

Entonces, es necesario configurar el archivo .bashrc, en cada uno de los nodos del clúster, para posteriormente establecer la configuración de ssh.

- Ingresar al fichero *.bashrc* para establecer las variables de entorno de Hadoop.

```
[root@master-node ~]# nano .bashrc
```

Figura 4.20: Abrir fichero .bashrc.

- Digitar los siguientes comandos en el fichero *.bashrc* para establecer las variables de entorno de Hadoop.

```
export HADOOP_HOME=/opt/hadoop
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_INSTALL=$HADOOP_HOME
```

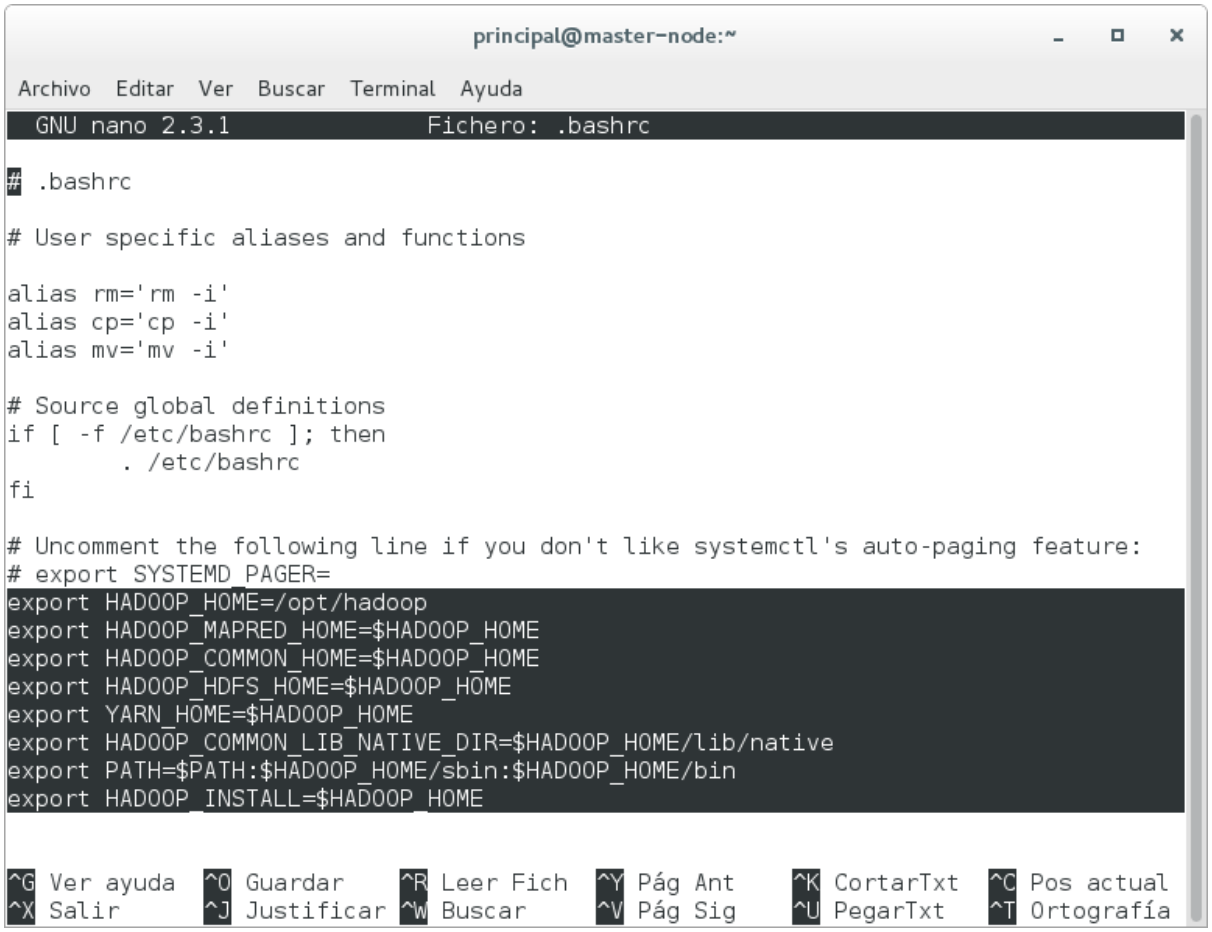


Figura 4.21: Configuración de variables de entorno de Hadoop en el fichero .bashrc.

- Iniciar la configuración *ssh* ejecutando el siguiente comando en la máquina desde la cual se va realizar el manejo remoto, en este caso se ejecutará el comando en la máquina correspondiente al nodo master, ya que desde ésta se va a controlar el manejo de los nodos esclavo.
- Al ejecutar el siguiente comando el sistema se encarga de establecer ciertas configuraciones, estas configuraciones serán detalladas a continuación.

```
#ssh-keygen –t rsa
```

- Establecer la dirección en donde se va a guardar esta configuración. Esta configuración debe ser realizada exclusivamente en la máquina que se conectará a los nodos esclavos.
- Posteriormente, se presentará en pantalla un mensaje que solicite el ingreso de una contraseña. No se debe ingresar ninguna clave, y solo presionar la tecla Enter.
- Finalmente se presenta un mensaje de confirmación de contraseña, se debe presionar Enter y así, la configuración *ssh* sin contraseña estará disponible.

- En esta sección se debe ejecutar el siguiente comando en cada uno de los nodos esclavos, para que la carpeta tenga la configuración *ssh.*

```
#mkdir ~/.ssh
```

- Finalmente, se introduce la clave pública del computador master en el fichero de llaves para la autorización del manejo remoto de los nodos esclavos. Al ejecutar este comando, se pedirá por última vez la contraseña del nodo master.

```
#cat .ssh/id_rsa.pub | ssh root@nodos_esclavos 'cat >> .ssh/authorized_keys'
```

- Para comprobar se debe digitar el siguiente comando, si se conecta sin contraseña la configuración fue realizada con éxito.

```
#ssh root@nodos_esclavos
```

Con estas configuraciones se pueden omitir pasos que contienen el encendido de los servicios y demonios de Hadoop en todos los nodos de manera manual, ya que con solo ejecutar el comando *start-all.sh* en la máquina del nodo master, se encienden automáticamente los servicios de este nodo y de los nodos esclavos.

PASO 3:

Creación de un Grupo de Usuarios

- Se crea el grupo Hadoop y los usuarios en donde se van a instalar los componentes de Hadoop. Este paso es importante realizarlo tanto en el Nodo Master como en los Nodos Esclavo.

Se digitan los siguientes comandos:

- Para crear el grupo hadoop:

```
#groupadd hadoop
```

```
[root@master-node ~]# groupadd hadoop
```

Figura 4.22: Creación grupo Hadoop.

- Para crear los usuarios que van a formar parte del grupo hadoop:

```
#useradd –g hadoop yarn
#useradd –g hadoop hdfs
#useradd –g hadoop mapred
```

```
[root@master-node ~]# useradd -g hadoop yarn
[root@master-node ~]# useradd -g hadoop hdfs
[root@master-node ~]# useradd -g hadoop mapred
[root@master-node ~]#
```

Figura 4.23: Creación de usuarios del grupo Hadoop.

PASO 4:

Creación de Directorios en el Nodo Master

- Previamente se deben tener guardados en la máquina correspondiente al Nodo Master los siguientes Scripts:

env-variable

```
#!/bin/bash

# Users and Groups
export HDFS_USER=hdfs
export YARN_USER=yarn
export MAPRED_USER=mapred
export HADOOP_GROUP=hadoop

# Hadoop Service - HDFS
export DFS_NAME_DIR='/var/data/hadoop/hdfs/nn'
export DFS_DATA_DIR='/var/data/hadoop/hdfs/dn'
export DFS_CHECKPOINT_DIR='/var/data/hadoop/hdfs/snn'
export DFS_LOG_DIR='/opt/hadoop/logs'

# Hadoop Service - YARN
export YARN_LOCAL_DIR='/var/hadoop/yarn/local'
export YARN_LOCAL_LOG_DIR='/var/hadoop/yarn/logs'
```

directories_master

```
#!/bin/bash

echo "Create namenode local dir"
mkdir -p $DFS_NAME_DIR;
chown -R $HDFS_USER:$HADOOP_GROUP $DFS_NAME_DIR;
chmod -R 755 $DFS_NAME_DIR;

echo "Create checkpoint dir"
mkdir -p $DFS_CHECKPOINT_DIR;
chown -R $HDFS_USER:$HADOOP_GROUP $DFS_CHECKPOINT_DIR;
chmod -R 755 $DFS_CHECKPOINT_DIR;

echo "Create hadoop logs dir"
mkdir -p $DFS_LOG_DIR;
chown -R $HDFS_USER:$HADOOP_GROUP $DFS_LOG_DIR;
chmod -R 775 $DFS_LOG_DIR;
```

Ambos Scripts deben ser guardados en el equipo con la extensión: ***.sh***

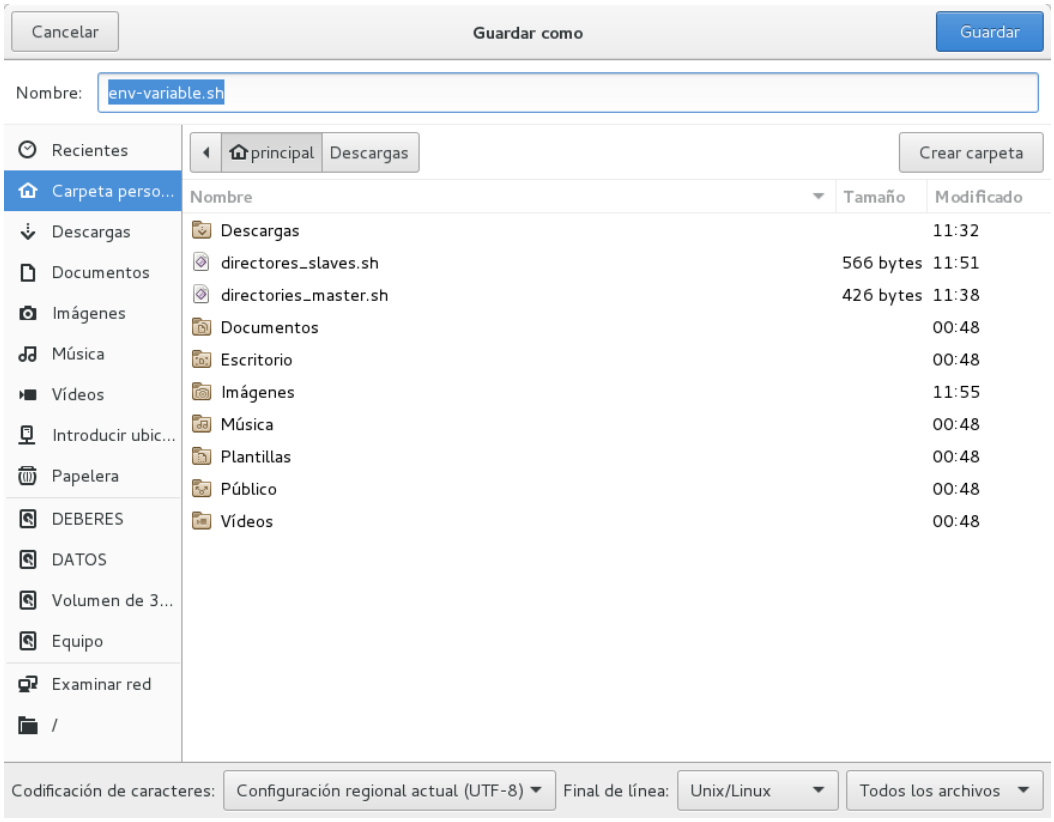


Figura 4.24: Guardar script env-variable.sh *en el equipo.*

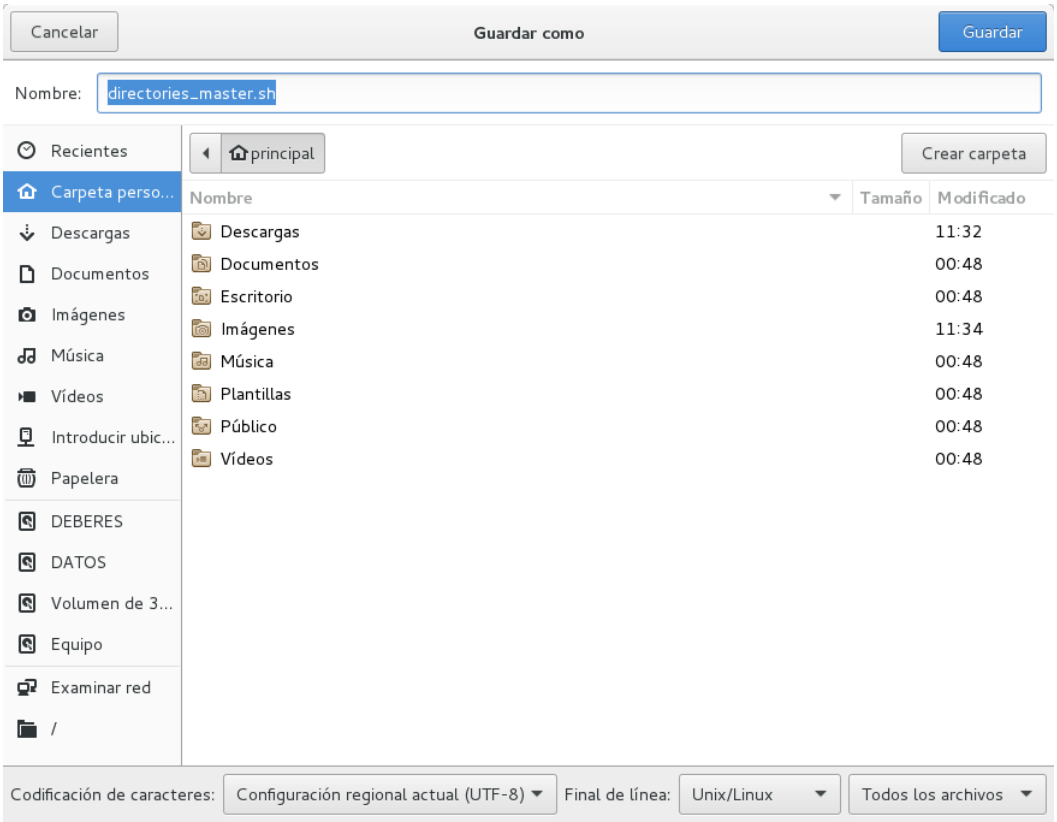


Figura 4.25: Guardar script directories_master.sh *en el equipo.*

❖ Copiar los escripts: *env-variable* y *directories_master* al directorio *root.*

Se digitan los siguientes comandos:

- Para copiar el script *env-variable* al directorio *root.*

```
#cp /home/principal/Descargas/install_hadoop_cluster/env-variable /root/
```

- Para copiar el script *directories_master* al directorio *root.*

```
#cp /home/principal/Descargas/install_hadoop_cluster/directories_master
/root/
```

```
[root@master-node ~]# cp /home/principal/Descargas/install_hadoop_cluster/env-variable
/root/
[root@master-node ~]# ls
anaconda-ks.cfg  hadoop-2.7.2.tar.gz   jdk-8u91-linux-x64.rpm
env-variable     initial-setup-ks.cfg
[root@master-node ~]# cp /home/principal/Descargas/install_hadoop_cluster/directories_m
aster /root/
[root@master-node ~]# ls
anaconda-ks.cfg     env-variable         initial-setup-ks.cfg
directories_master  hadoop-2.7.2.tar.gz  jdk-8u91-linux-x64.rpm
[root@master-node ~]#
```

Figura 4.26: Copiar scripts env-variable *y* directories_master *al directorio* root.

❖ Dar permisos para poder ejecutar los Scripts: *env-variable* y *directories_master*

```
#chmod 775 env-variable directories_master
```

```
[root@master-node ~]# chmod 755 env-variable directories_master
```

Figura 4.27: Permisos para ejecutar los scripts: env-variable *y* directories_master.

❖ Verificar si los scripts se activaron correctamente:

```
#ls
```

```
[root@master-node ~]# ls
anaconda-ks.cfg     env-variable  hadoop-2.7.2.tar.gz   jdk-8u91-linux-x64.rpm
directories_master  hadoop-2.7.2  initial-setup-ks.cfg
```

Figura 4. 28: Comprobación que los scripts se activaron correctamente.

❖ De ser así, ejecutar cada uno de los Scripts y crear los directorios que se muestran:

```
#. env-variable
#./directories_master
```

```
[root@master-node ~]# . env-variable
[root@master-node ~]# ./directories_master
Create namenode local dir
Create checkpoint dir
Create hadoop logs dir_
```

Figura 4. 29: Ejecución de los scripts y creación de directorios.

Creación de Directorios en los Nodos Esclavo

- Al igual que la creación de directorios en el Nodo Master, para los Nodos Esclavo se deben tener previamente guardados en las máquinas correspondientes a estos nodos los siguientes Scripts:

env-variable

```
#!/bin/bash

# Users and Groups
export HDFS_USER=hdfs
export YARN_USER=yarn
export MAPRED_USER=mapred
export HADOOP_GROUP=hadoop

# Hadoop Service - HDFS
export DFS_NAME_DIR='/var/data/hadoop/hdfs/nn'
export DFS_DATA_DIR='/var/data/hadoop/hdfs/dn'
export DFS_CHECKPOINT_DIR='/var/data/hadoop/hdfs/snn'
export DFS_LOG_DIR='/opt/hadoop/logs'

# Hadoop Service - YARN
export YARN_LOCAL_DIR='/var/hadoop/yarn/local'
export YARN_LOCAL_LOG_DIR='/var/hadoop/yarn/logs'
```

directories_slaves

```
#!/bin/bash

echo "Create datanode local dir"
mkdir -p $DFS_DATA_DIR;
chown -R $HDFS_USER:$HADOOP_GROUP $DFS_DATA_DIR;
chmod -R 750 $DFS_DATA_DIR;
```

```
echo "Create yarn local dir"
mkdir -p $YARN_LOCAL_DIR;
chown -R $YARN_USER:$HADOOP_GROUP $YARN_LOCAL_DIR;
chmod -R 755 $YARN_LOCAL_DIR;

echo "Create yarn local log dir"
mkdir -p $YARN_LOCAL_LOG_DIR;
chown -R $YARN_USER:$HADOOP_GROUP $YARN_LOCAL_LOG_DIR;
chmod -R 755 $YARN_LOCAL_LOG_DIR;

echo "Create hadoop log dir"
mkdir -p $DFS_LOG_DIR;
chown -R $HDFS_USER:$HADOOP_GROUP $DFS_LOG_DIR;
chmod -R 775 $DFS_LOG_DIR;
```

Estos Scripts deben ser guardados con la extensión: ***.sh***

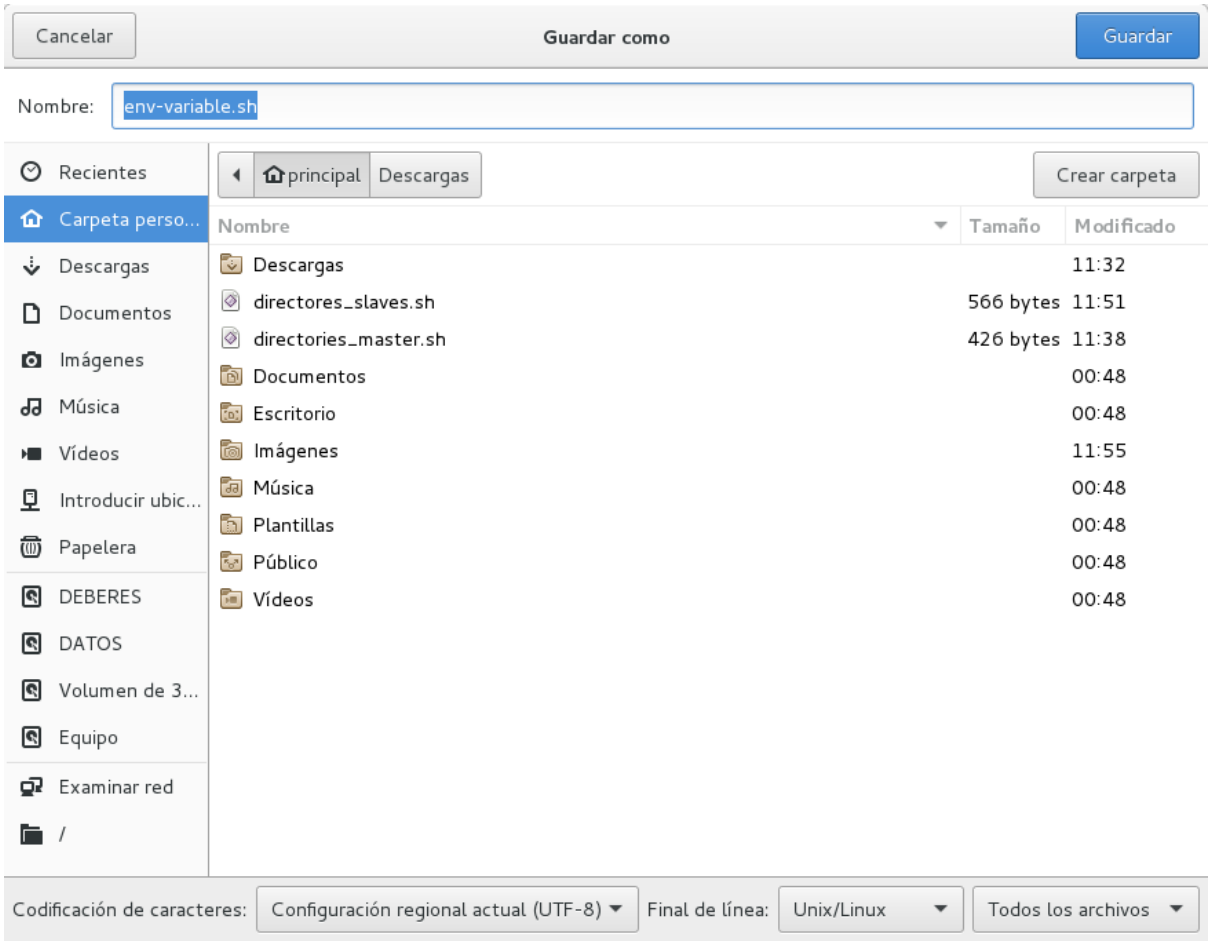


Figura 4.30: Guardar script env-variable.sh *en el equipo.*

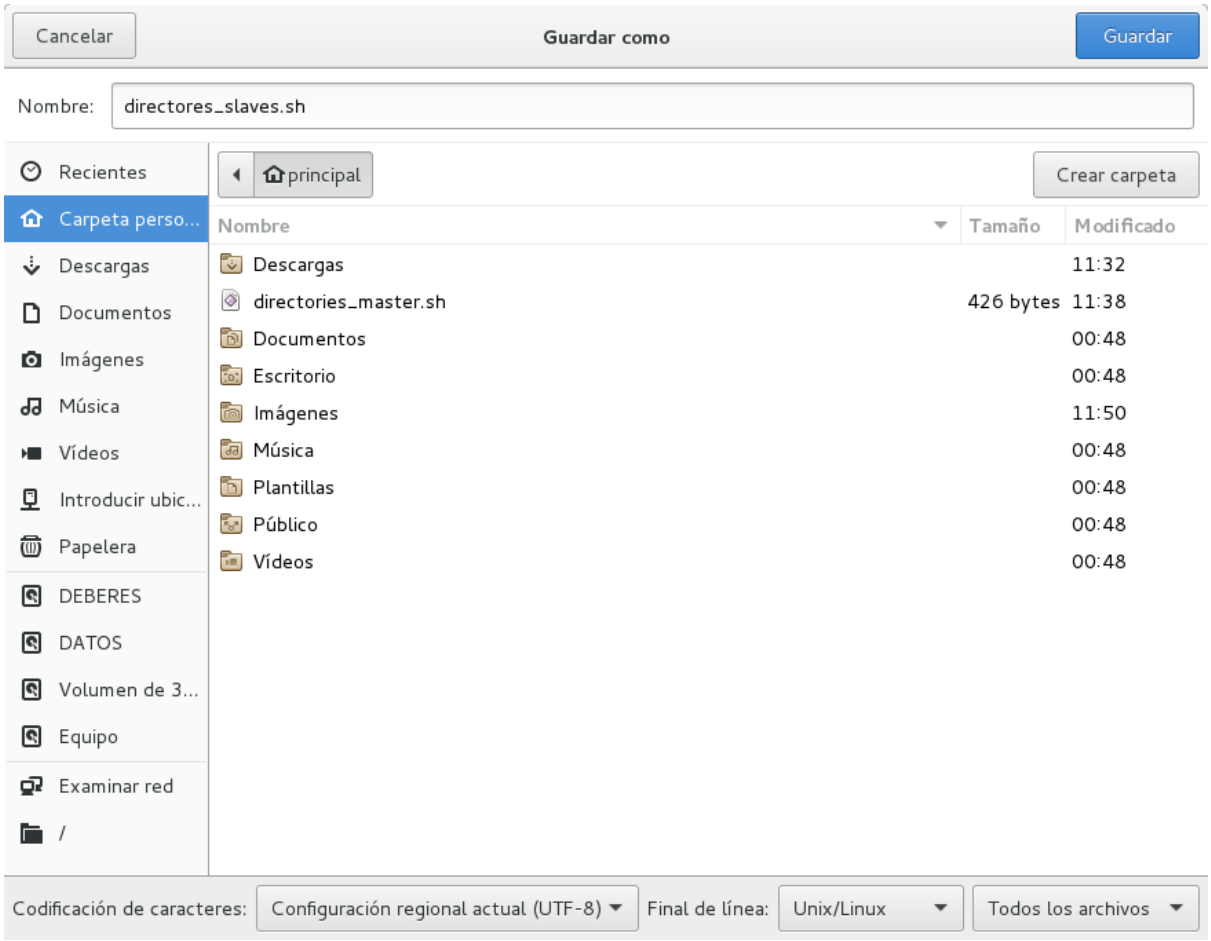


Figura 4.31: Guardar script directories_slaves.sh *en el equipo.*

- ❖ Copiar los escripts: *env-variable* y *directories_slaves* al directorio *root.*

Se digitan los siguientes comandos:

- Para copiar el script *env-variable* al directorio *root.*

```
#cp /home/principal/Descargas/install_hadoop_cluster/env-variable /root/
```

- Para copiar el script *directories_slaves* al directorio *root.*

```
#cp /home/principal/Descargas/install_hadoop_cluster/directories_slaves
/root/
```

```
[root@slave1-node ~]# cp /home/secundario1/Descargas/install_hadoop_cluster/env-
variable /root/
[root@slave1-node ~]# ls
anaconda-ks.cfg  hadoop-2.7.2.tar.gz   jdk-8u91-linux-x64.rpm
env-variable     initial-setup-ks.cfg
[root@slave1-node ~]# cp /home/secundario1/Descargas/install_hadoop_cluster/dire
ctories_slaves /root/
[root@slave1-node ~]# ls
anaconda-ks.cfg     env-variable         initial-setup-ks.cfg
directories_slaves  hadoop-2.7.2.tar.gz  jdk-8u91-linux-x64.rpm
[root@slave1-node ~]#
```

Figura 4.32: Copiar scripts env-variable *y* directories_slaves *al directorio* root.

- Dar permisos para poder ejecutar los Scripts: *env-variable* y *directories_slaves:*

```
#chmod 775 env-variable directories_slaves
```

```
[root@slave1-node ~]# chmod 755 env-variable directories_slaves
```

Figura 4.33: Permisos para Ejecutar los scripts: env-variable *y* directories_slaves.

- Verificar si los scripts se activaron correctamente en el nodo correspondiente:

```
#ls
```

```
[root@slave1-node ~]# ls
anaconda-ks.cfg     env-variable  hadoop-2.7.2.tar.gz   jdk-8u91-linux-x64.rpm
directories_slaves  hadoop-2.7.2  initial-setup-ks.cfg
```

Figura 4.34: Comprobación que los scripts se activaron correctamente.

- De ser así, ejecutar los Scripts y crear los directorios que se muestran:

```
#. env-variable
#./directories_slaves
```

```
[root@slave1-node slave1-node]# . env-variable
[root@slave1-node slave1-node]# ./directories_slaves
Create datanode local dir
Create yarn local dir
Create yarn local log dir
Create hadoop log dir
```

Figura 4.35: Ejecución de los scripts y creación de directorios.

Antes de continuar con los pasos subsiguientes es necesario explicar algunos conceptos que son necesarios para comprender como se debe realizar la configuración de Hadoop.

En la arquitectura de Hadoop existen una serie de bloques o demonios que son importantes para gestionar las actividades y replicaciones de los archivos HDFS (*Hadoop Distributed File System – Sistema de Archivos Distribuidos de Hadoop),* dentro de un clúster de hadoop. Dependiendo de la distribución de Hadoop que se desee ejecutar existen diferentes demonios, pero en la presente configuración los demonios que se ejecutarán son los siguientes: Name Node, Data Node, Secondary Name Node, Job History Server, Resource Manager y Node Manager.

En este caso, el Name Node, el Secondary Name Node, Job History Server, y Resource Manager serán ejecutados en una sola máquina correspondiente al Nodo Master, pero en grupos de producción que cuentan con más de 20 nodos aproximadamente, estos demonios pueden ser ejecutados en nodos separados. Mientras que los demonios Data Node y Node Manager serán ejecutados en las máquinas correspondientes a los Nodos Esclavo.

Para ello es necesario conocer en qué consisten cada uno de estos demonios.

Name Node: El Name Node en Hadoop es el nodo donde Hadoop almacena toda la información de la ubicación de los archivos HDFS. En otras palabras, mantiene los metadatos de HDFS. Es un punto único de fallo para el clúster Hadoop. Sin este servicio, no hay manera de acceder a los archivos HDFS. El Name Node no almacena los datos reales, los propios datos se almacenan en los Data Nodes.

Data Node: El Data Node se encarga de almacenar los datos reales en HDFS. Gestiona los bloques de archivos dentro del nodo. Se envía información al *Name Node* acerca de los archivos y los bloques almacenados en ese nodo y responde al *Name Node* para todas las operaciones del sistema de archivos, es decir, el Name Node y el o los Data Nodes están en constante comunicación entre ellos.

Secondary Name Node: El Secondary Name Node es el responsable de realizar funciones de mantenimiento periódicas para el *Name Node*. Sólo crea puntos de comprobación del sistema de ficheros presentes en el Name Node.

Job History Server: El Job History Server puede ser ejecutado de manera autónoma en un nodo dentro del clúster de Hadoop o dentro del Nodo Master, pero se recomienda ser ejecutado en el mismo nodo en cual se corra el Resource Manager. Su principal función es la de almacenar y mantener un historial de los trabajos de MapReduce que se ejecuten en el clúster. En versiones inferiores de Hadoop (Hadoop-2.0.0), el Job History Server funcionaba

como Job Tracker, pero éste en versiones nuevas como la que se está utilizando para la presente configuración (Hadoop-2.7.2) ya no existe debido a los errores que presentaba en el manejo de los historiales de MapReduce.

Resource Manager: El Resource Manager es el componente central del YARN y regula todos los recursos de procesamiento de datos dentro del clúster Hadoop, es decir, el Resource Manager es un planificador dedicado que asigna recursos a las aplicaciones que se solicitan. Sus tareas son sólo para mantener una visión global de todos los recursos del clúster, el manejo de las solicitudes de recursos, la programación de la solicitud, y luego la asignación de recursos a la aplicación solicitante. El Resource Manager es un componente crítico en un clúster Hadoop, debe ejecutarse en un nodo maestro dedicado.

Node Manager: Cada nodo esclavo tiene un demonio Node Manager, que actúa como un esclavo para el Resource Manager. Cada nodo esclavo tiene un servicio que lo ata al servicio de procesamiento (Node Manager) y al servicio de almacenamiento (DataNode) que permiten a Hadoop ser un sistema distribuido. Cada Node Manager registra los recursos de procesamiento de datos disponibles en el nodo esclavo y envía informes periódicos al Resource Manager.

Por otro lado, existen archivos de configuración de los componentes de Hadoop que son importantes para su correcto funcionamiento, estos son: el core-site.xml, hdfs-site.xml, mapred-site.xml y yarn-site.xml.

El archivo core-site.xml, contiene opciones de configuración que son comunes para todos los servidores de la agrupación, es decir, en este archivo se establecen el nombre del sistema de archivos y el puerto a través del cual se recibirán las peticiones del cliente, se recomienda colocar el puerto 9000, para que se carguen correctamente las configuraciones de hadoop en el servidor web.

El archivo hdfs-site.xml proporciona la configuración de los archivos HDFS (*Hadoop Distributed File System – Sistema de Archivos Distribuidos de Hadoop), es decir, en éste se definen los* directorios que van a ser usados como **Name Node y Data Node, además se establece** el factor de replicación de datos, que se refiere al número de Nodos Esclavo que se utilizarán dentro del cluster de Hadoop.

El archivo mapred-site.xml, es importante para establecer el framework que se va a utilizar para procesar las aplicaciones MapReduce. En esta instalación se utilizará el framework

YARN debido a que es el encargado de ejecutar los procesos de las aplicaciones que están desarrolladas en MapReduce.

El archivo yarn-site.xml indica el servicio de intercambio (shuffle) que es necesario para ejecutar las aplicaciones MapReduce.

Finalmente, es preciso comprender lo que significa el YARN dentro de hadooop debido que esta configuración será realizada en base a este framework.

YARN: El Yarn es un administrador de recursos que se creó mediante la separación de las capacidades del motor de procesamiento y gestión de recursos de MapReduce que se implementó en la versión 1.0 de Hadoop. Yarn está presente a partir de la versión 2.1 de Hadoop y es a menudo llamado el sistema operativo de Hadoop, ya que es responsable de la gestión y el seguimiento de las cargas de trabajo, el mantenimiento de un multi-entorno de distribución, la implementación de controles de seguridad, y la gestión de funciones de alta disponibilidad de Hadoop. Al igual que un sistema operativo en el servidor, Yarn está diseñado para permitir a diversas aplicaciones de usuario que se ejecutan en una plataforma multi-nodo. (Sullivan, 2014)

En seguida, se continúan con los pasos de instalación necesarios para ejecutar Hadoop.

PASO 5:

Editar el Archivo *core-site.xml* en el Nodo Master

- Se abre el archivo *core-site.xml* digitando lo siguiente:

```
#nano /opt/hadoop/etc/hadoop/core-site.xml
```

```
[root@master-node ~]# nano /opt/hadoop/etc/hadoop/core-site.xml
```

Figura 4.36: Abrir archivo core-site.xml *en el Nodo Master.*

Se abre el fichero correspondiente al *core-site.xml* como se muestra a continuación:

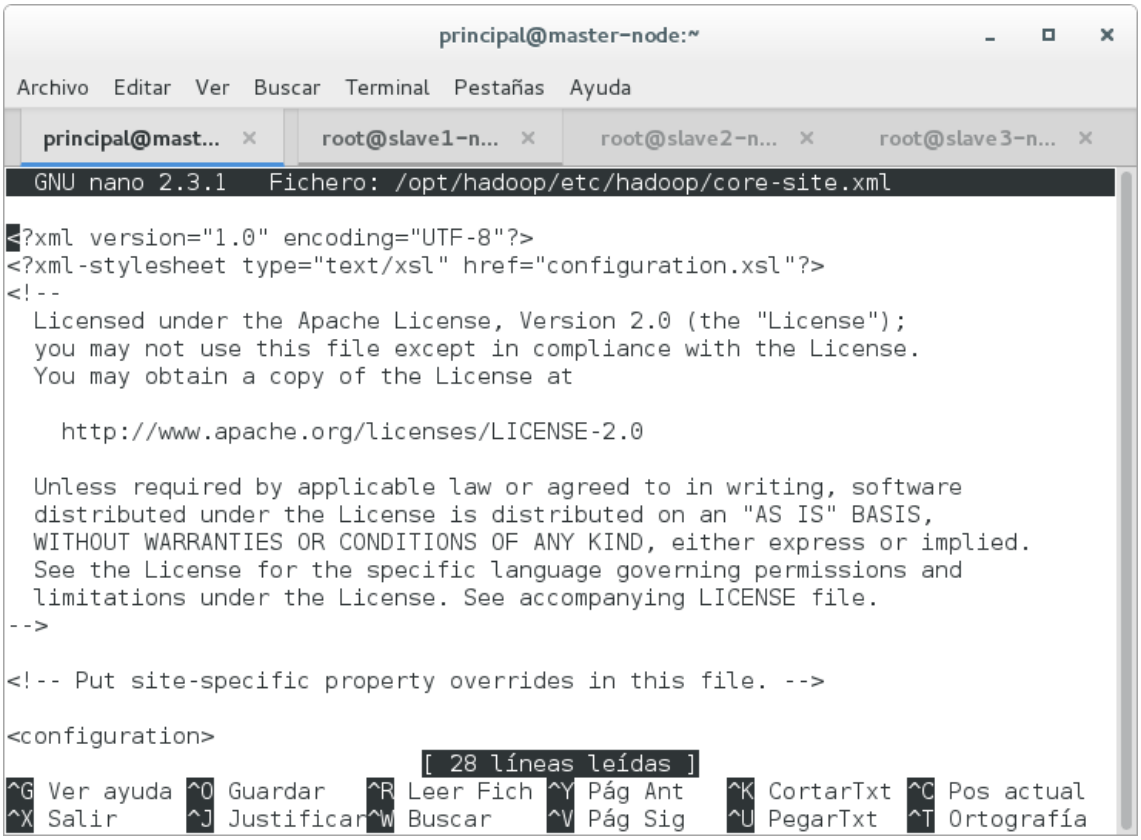


Figura 4.37: Archivo core-site.xml *Nodo Master.*

- Ubicarse en la siguiente línea de código, `<configuration>`

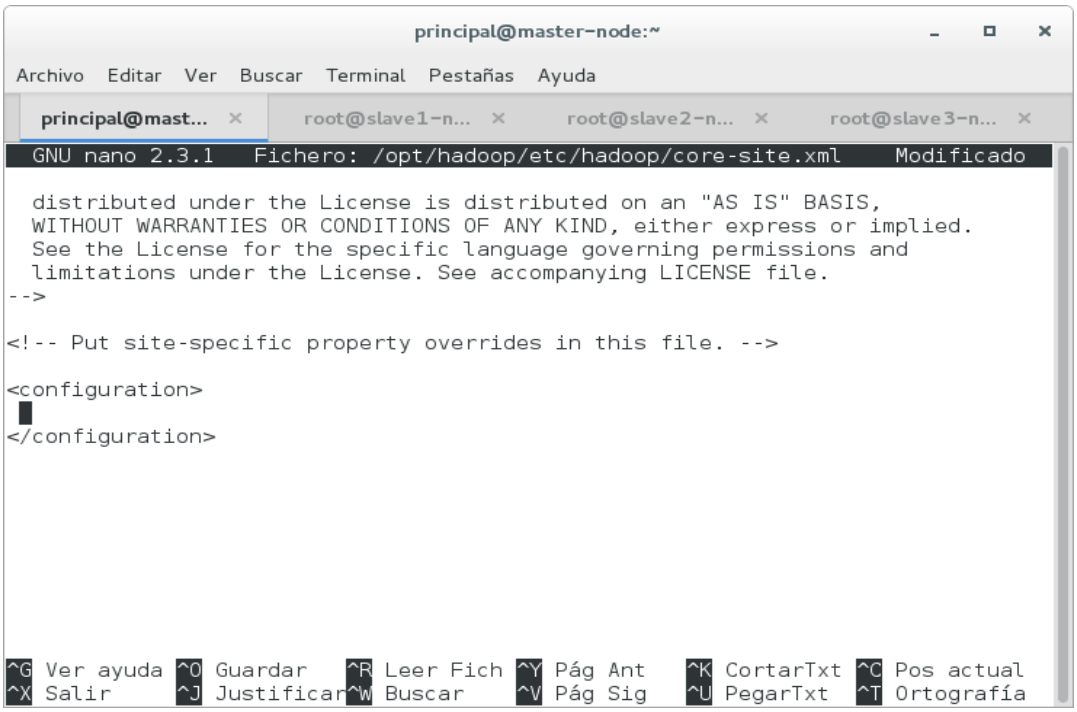


Figura 4.38: Archivo core-site.xml *Nodo Master.*

- Colocar las siguientes líneas de código como se muestra a continuación:

core-site.xml

```
1. <configuration>
2. <property>
3. <name>fs.defaultFS</name>
4. <value>hdfs://master-node.centos:9000</value>
5. </property>
6. <property>
7. <name>hadoop.http.staticuser.user</name>
8. <value>hdfs</value>
9. </property>
10.      </configuration>
```

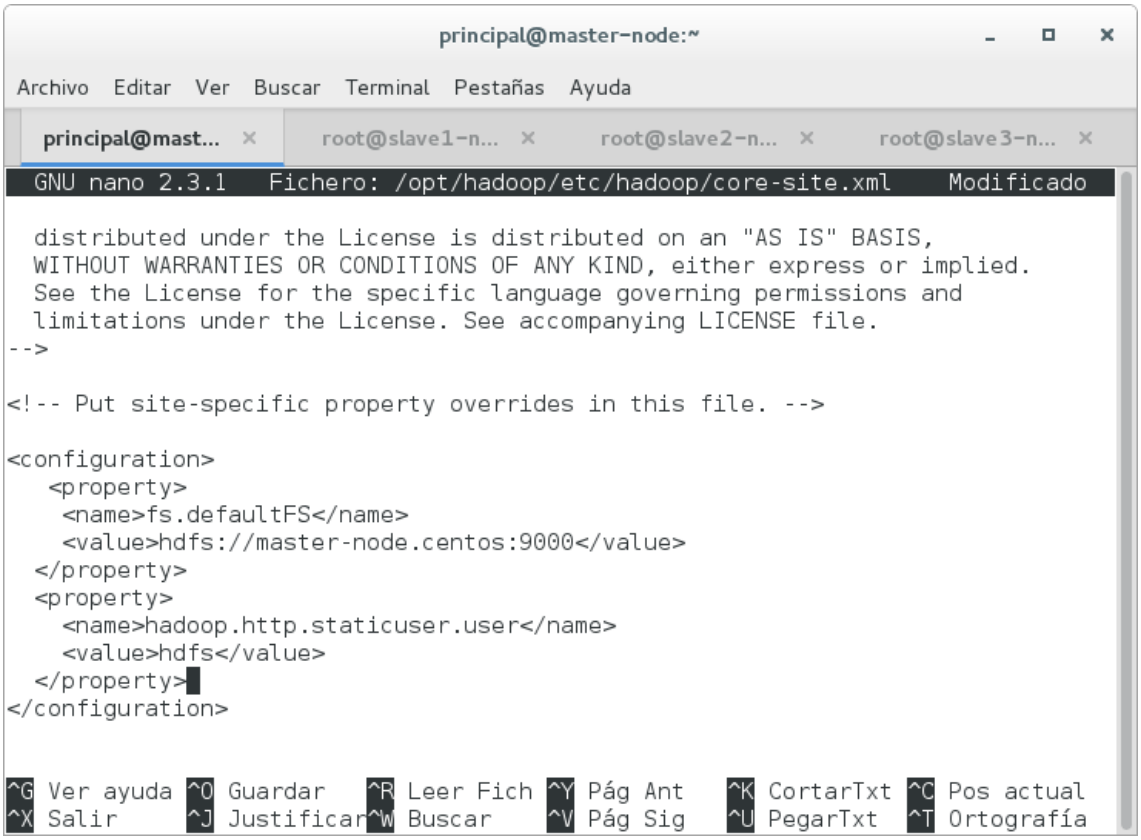


Figura 4.39: Agregación de líneas de código al archivo core-site.xml *Nodo Master.*

Presionar Ctrl + X (para salir)

Como se puede observar en el código escrito en la parte superior, la línea 4 está marcada con un color celeste: `<value>hdfs://master-node.centos:8020</value>`, esto es

necesario para resaltar que en esta línea de código se debe colocar el nombre de la máquina en la que está trabajando el Nodo Master, es decir, `master-node`, seguido del nombre del dominio que en este caso es `centos`, de esta manera la identificación completa de la máquina en la cual se está ejecutando el Nodo Master es: `master-node.centos`. Se establece además, el número del puerto del Name Node que hace referencia al Nodo Master y corresponde al puerto `9000`.

Editar el Archivo *core-site.xml* en los Nodos Esclavo

- Se abre el archivo *core-site.xml* para poder editarlo. Realizar estos pasos en cada uno de los nodos esclavo.

```
#nano /opt/hadoop/etc/hadoop/core-site.xml
```

```
[root@slave1-node ~]# nano /opt/hadoop/etc/hadoop/core-site.xml
```

Figura 4.40: Abrir archivo core-site.xml *Nodos Esclavo.*

Así, se abre el fichero correspondiente al *core-site.xml* en el Nodo Esclavo:

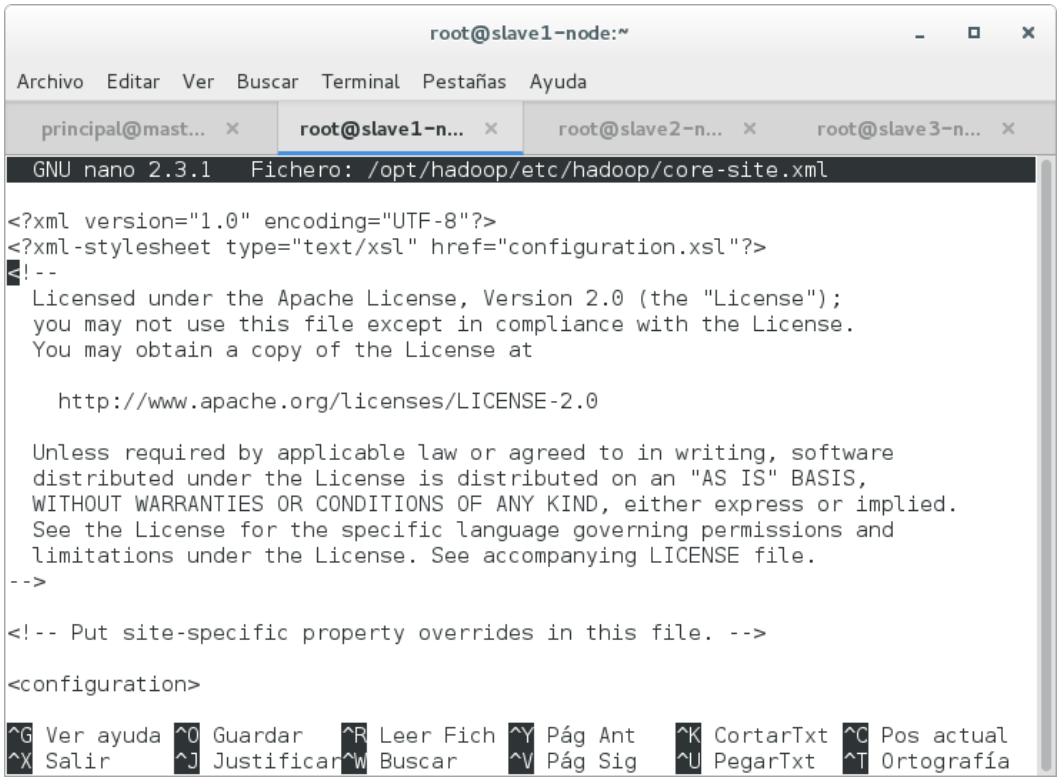


Figura 4.41: Archivo core-site.xml Nodos Esclavo.

- Ubicarse en la siguiente línea de código, `<configuration>`

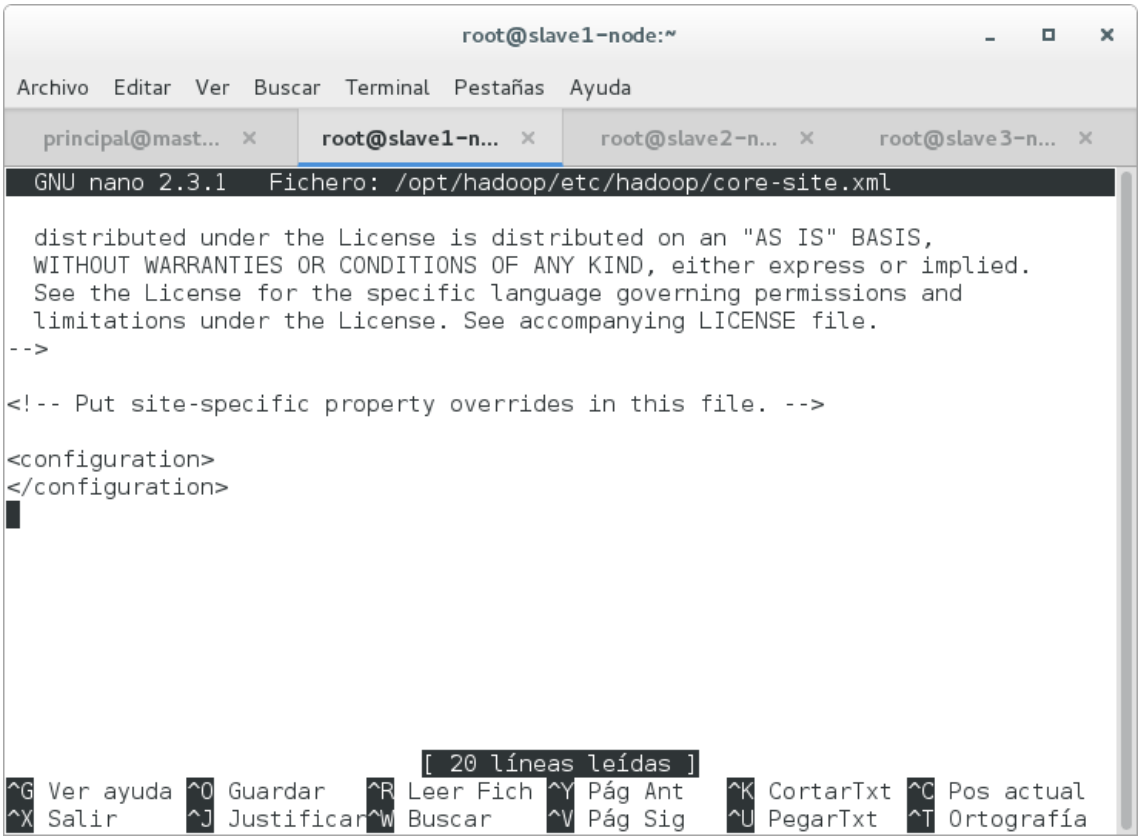


Figura 4.42: Archivo core-site.xml *Nodos Esclavo.*

- Colocar las siguientes líneas de código como se muestra a continuación:

core-site.xml

```
1. <configuration>
2. <property>
3. <name>fs.defaultFS</name>
4. <value>hdfs://master-node.centos:9000</value>
5. </property>
6. <property>
7. <name>hadoop.http.staticuser.user</name>
8. <value>hdfs</value>
9. </property>
10.      </configuration>
```

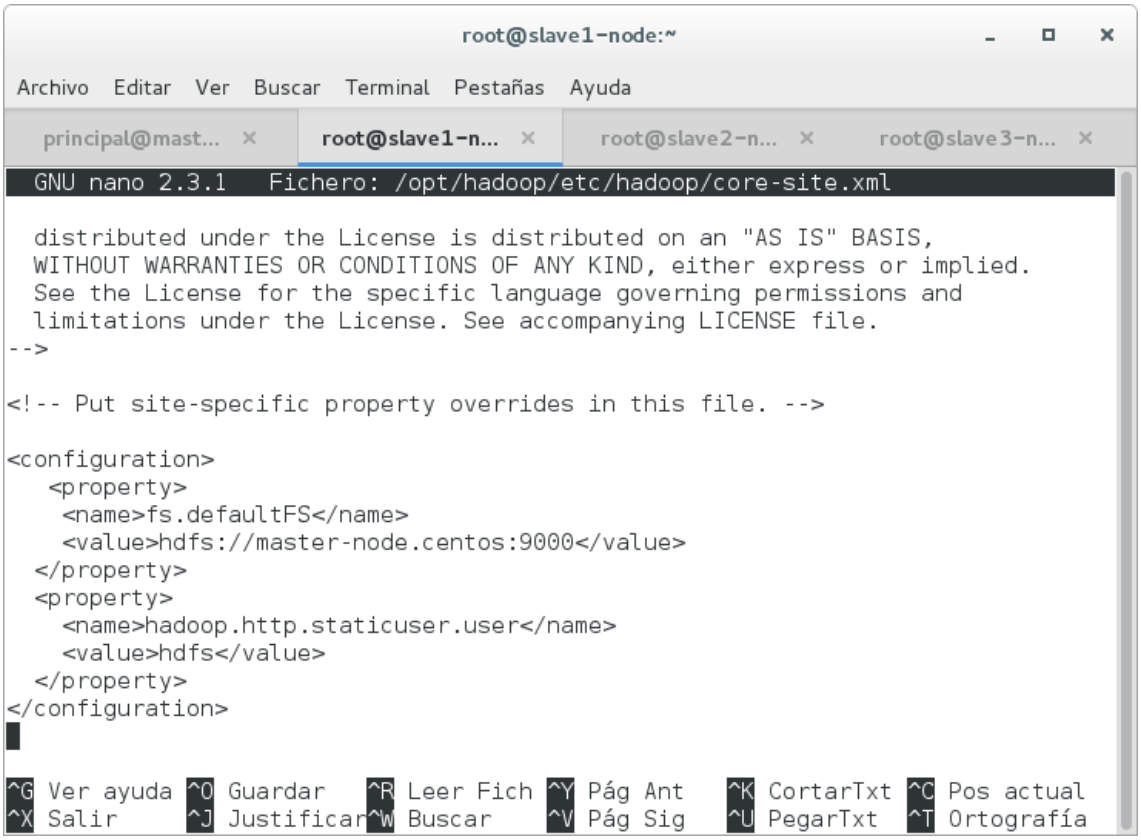


Figura 4.43: Agregación de líneas de código al archivo core-site.xml *Nodos Esclavo.*

Presionar Ctrl + X (para salir)

Como se puede observar en la parte superior la línea 4 está marcada de color celeste de esta manera: `<value>hdfs://master-node.centos:9000</value>`, en este caso se deben realizar las mismas configuraciones que se efectuaron en el archivo *core-site.xml* del Nodo Master, es decir, se debe colocar la identificación completa de la máquina en la cual está corriendo el Nodo Master, `master-node.centos`, y el puerto del Name Node que es el `9000.`

PASO 6:

Configuración del Archivo *mapred-site.xml* en el Nodo Master

- Se crea una copia del template del archivo *mapred-site.xml* que viene por defecto en hadoop:

```
#cp /opt/hadoop/etc/hadoop/mapred-site.xml.template
/opt/hadoop/etc/hadoop/mapred-site.xml
```

```
[root@master-node ~]# cp /opt/hadoop/etc/hadoop/mapred-site.xml.template /opt/ha
doop/etc/hadoop/mapred-site.xml
```

Figura 4.44: Copiar el Template mapred-site.xml *al directorio donde se encuentra Hadoop.*

- Se abre el archivo *mapred-site.xml*:

```
#nano /opt/hadoop/etc/hadoop/mapred-site.xml
```

```
[root@master-node ~]# nano /opt/hadoop/etc/hadoop/mapred-site.xml
```

Figura 4.45: Abrir el archivo mapred-site.xml *Nodo Master.*

- Se debe visualizar el siguiente archivo:

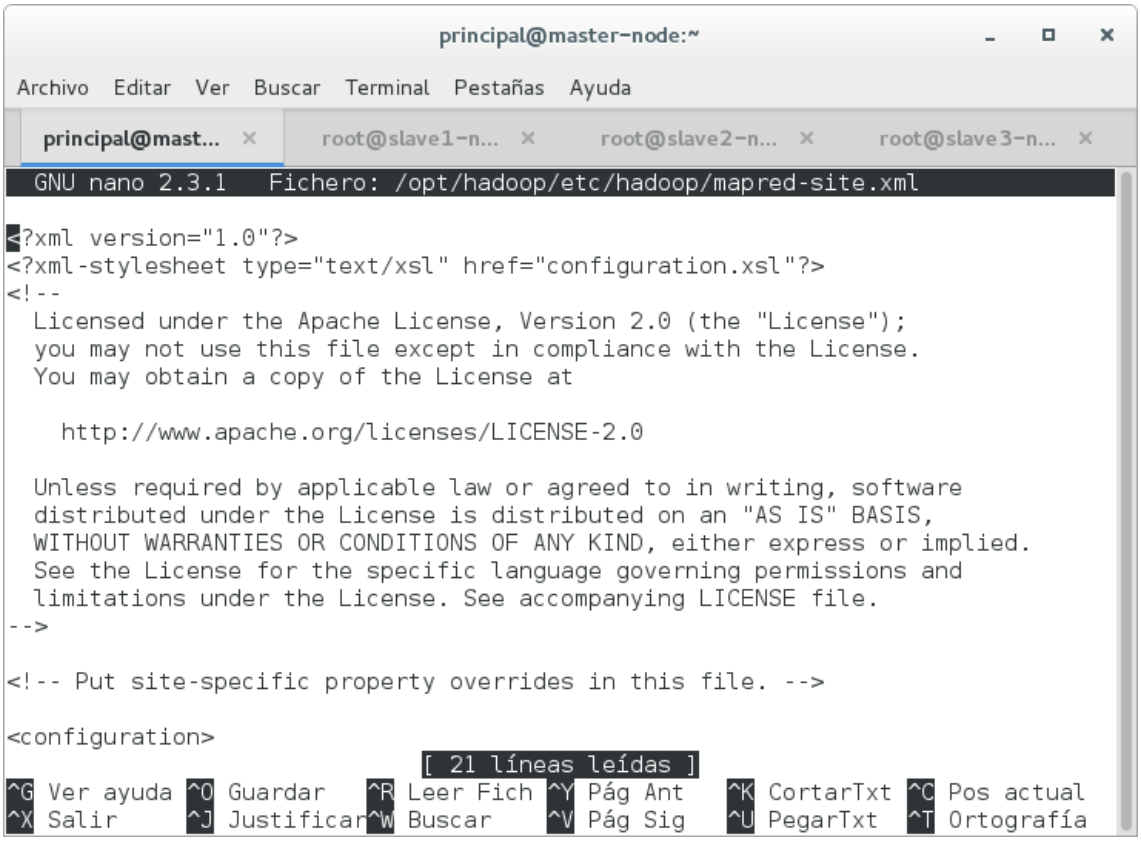


Figura 4.46: Archivo mapred-site.xml *Nodo Master.*

- Ubicarse en la siguiente línea de código: `<configuration>`:

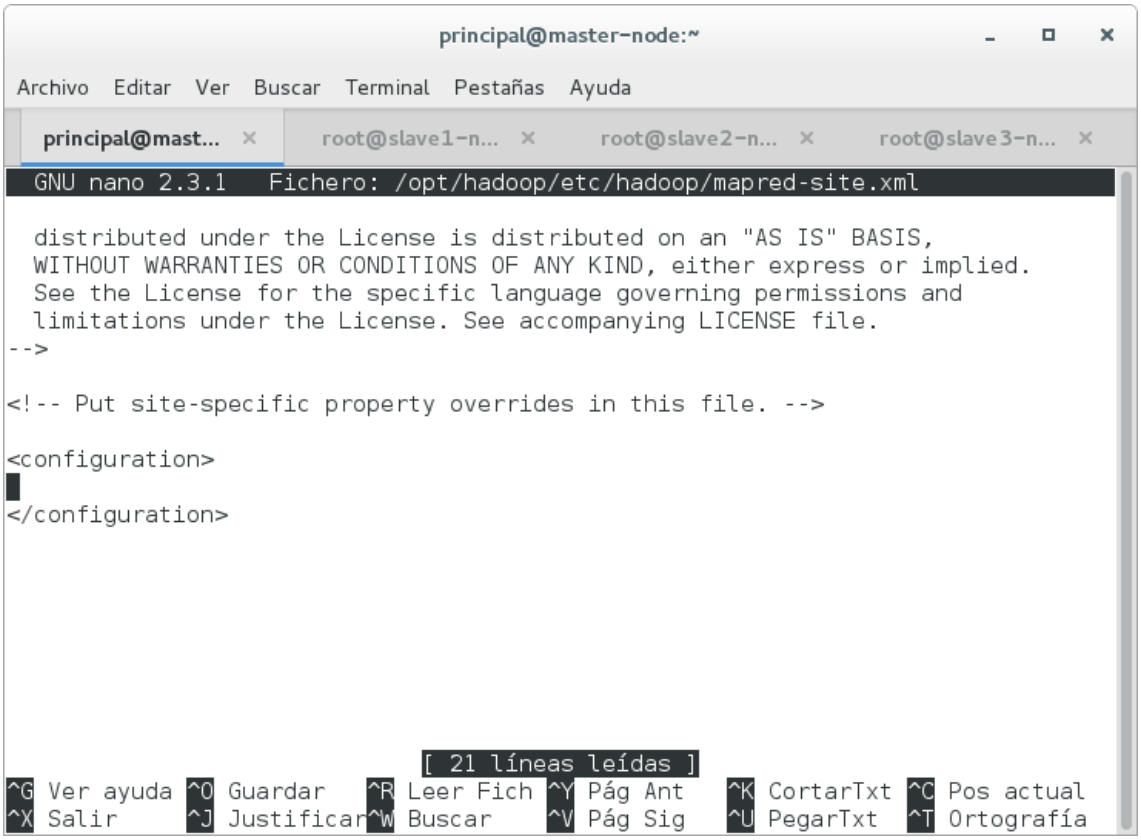


Figura 4.47: Archivo mapred-site.xml Nodo Master.

- Se define el nombre del framework que se utilizará para ejecutar MapReduce, que en este caso es YARN.

mapred-site.xml

```
1. <configuration>
2. <property>
3. <name>mapreduce.framework.name</name>
4. <value>yarn</value>
5. </property>
6. </configuration>
```

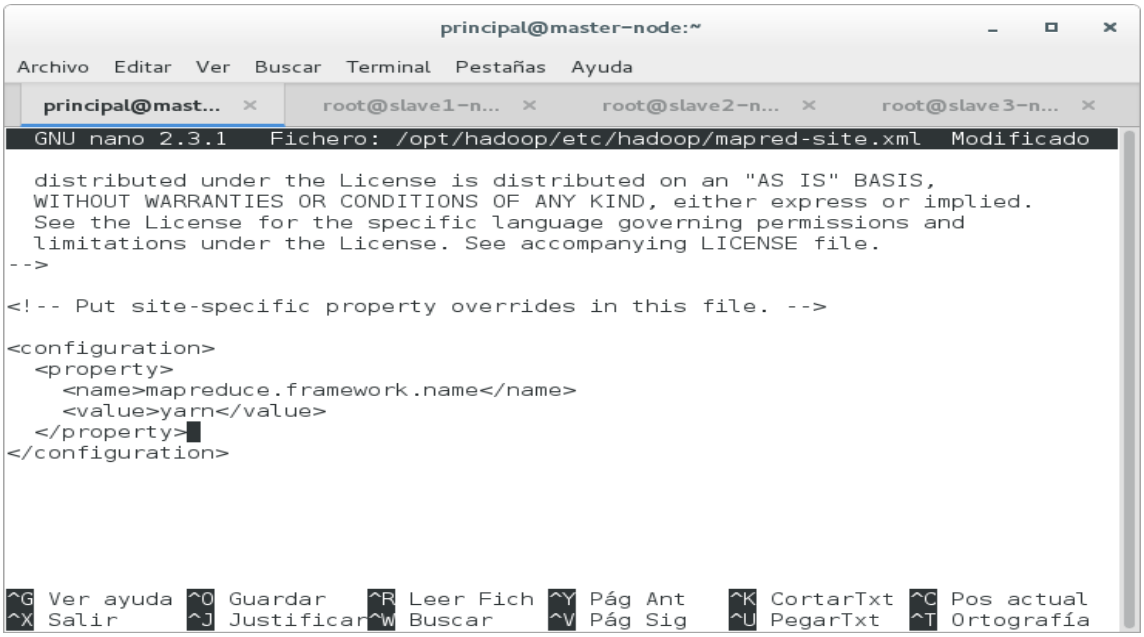


Figura 4.48: Agregación de líneas de código al archivo mapred-site.xml Nodo Master.

Presionar Ctrl + X (para salir)

Como se puede observar en la parte superior, la siguiente línea: `4. <value>yarn</value>`, establece el framework que es utilizado para ejecutar los componentes de MapReduce, es decir, YARN.

Configuración del Archivo *mapred-site.xml* en los Nodos Esclavo

- Se realiza un copia del template del archivo *mapred-site.xml*, tal como se lo hizo en el Nodo Master:

```
#cp /opt/hadoop/etc/hadoop/mapred-site.xml.template
/opt/hadoop/etc/hadoop/mapred-site.xml
```

```
[root@slave1-node ~]# cp /opt/hadoop/etc/hadoop/mapred-site.xml.template /opt/ha
doop/etc/hadoop/mapred-site.xml
```

Figura 4.49: Copiar el template del archivo mapred-site.xml *al directorio donde se encuentra Hadoop.*

- Se abre el archivo *mapred-site.xml*:

```
#nano /opt/hadoop/etc/hadoop/mapred-site.xml
```

```
[root@slave1-node ~]# nano /opt/hadoop/etc/hadoop/mapred-site.xml
```

Figura 4.50: Abrir archivo mapred-site.xml *en los Nodos Esclavos.*

- Se debe visualizar el siguiente archivo:

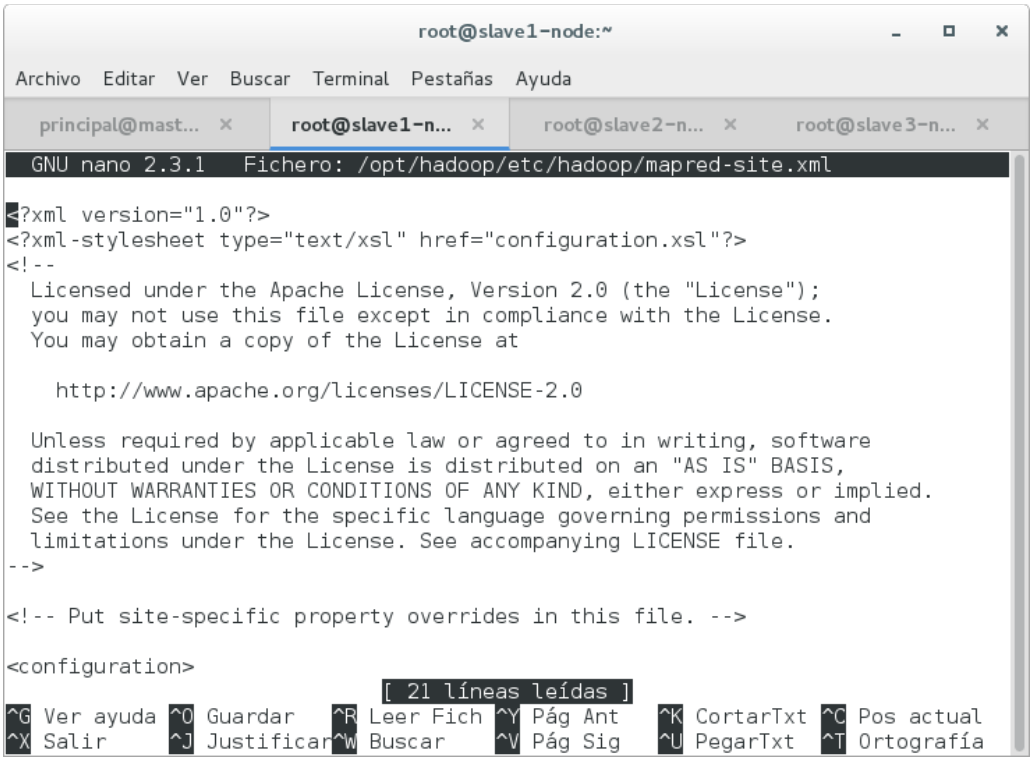


Figura 4.51: Archivo mapred-site.xml *en los Nodos Esclavos.*

- Ubicarse en la siguiente línea de código: `<configuration>`:

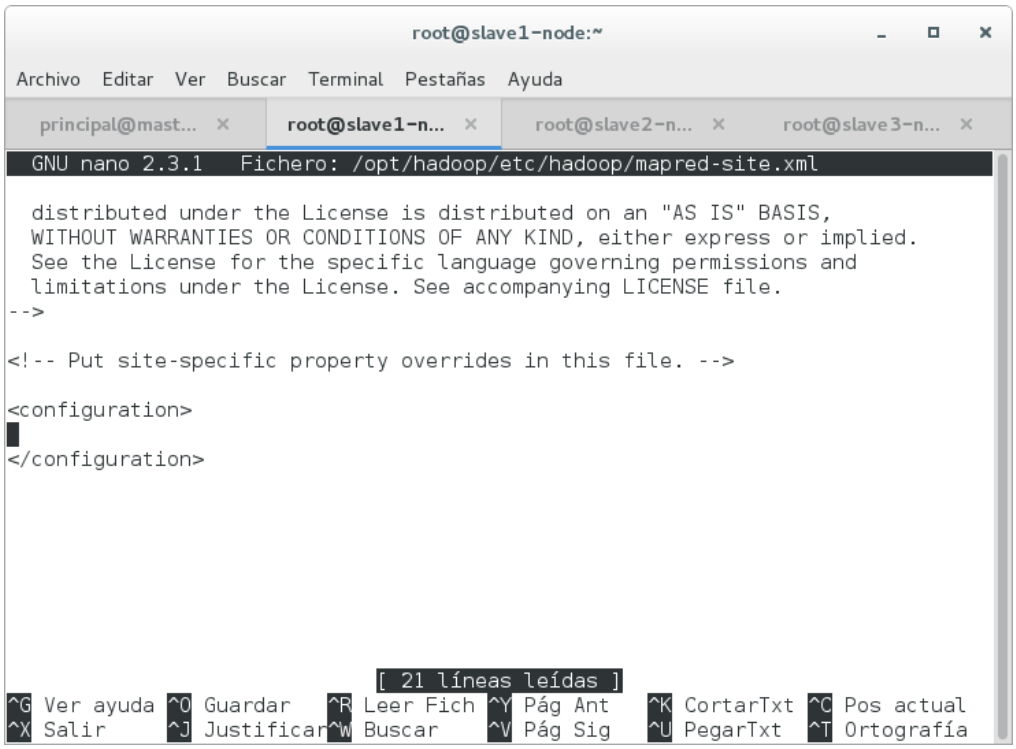


Figura 4. 52: Archivo mapred-site.xml *en los Nodos Esclavos.*

- Se copian en todos los Nodos Esclavo las mismas líneas de código que fueron utilizadas en la configuración del archivo *mapred-site.xml* del Nodo Master, puesto que el framework sigue siendo el mismo, es decir, YARN.

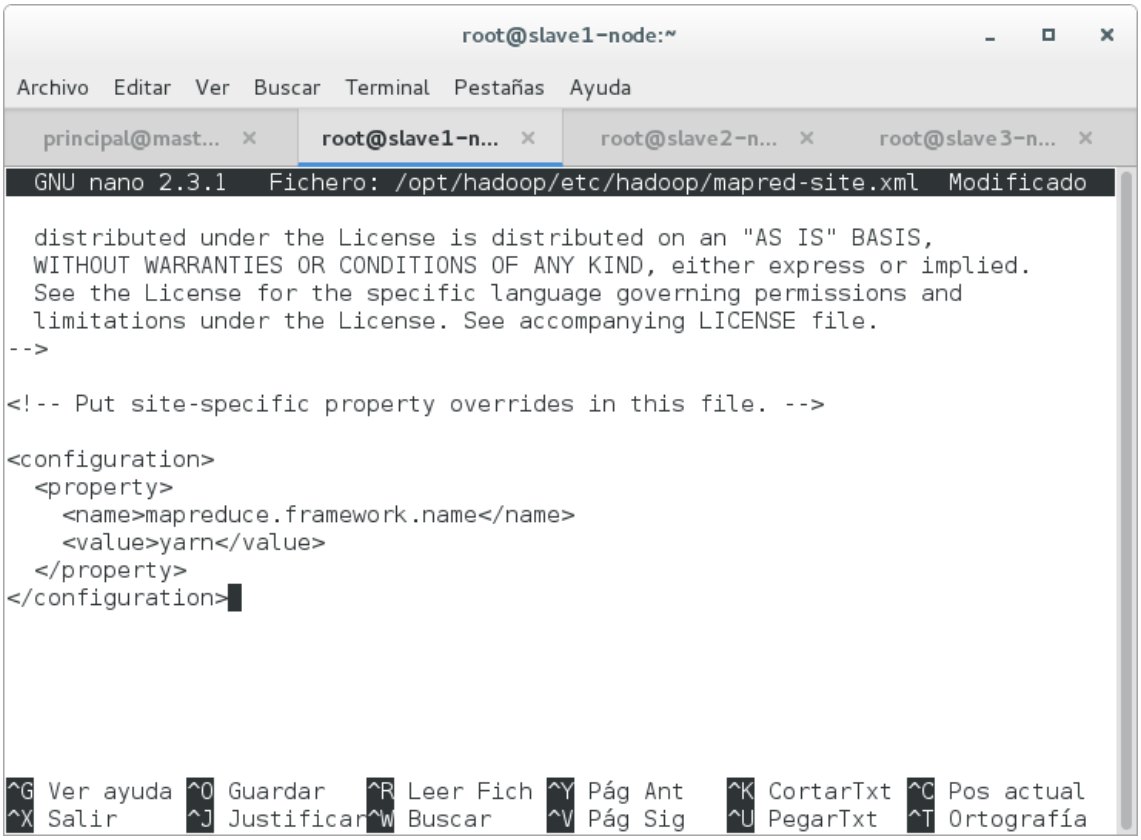


Figura 4.53: Agregación de líneas de código en el archivo mapred-site.xml en los Nodos Esclavos.

Presionar Ctrl + X (para salir)

PASO 7:

Configuración del archivo *hdfs-site.xml* en el Nodo Master:

- Se abre el *archivo hdfs-site.xml* digitando lo siguiente:

```
#nano /opt/hadoop/etc/hadoop/hdfs-site.xml
```

```
[root@master-node ~]# nano /opt/hadoop/etc/hadoop/hdfs-site.xml
```

Figura 4.54: Abrir archivo hdfs-site.xml *Nodo Master.*

- Se abre el siguiente fichero correspondiente al *hdfs-site.xml*:

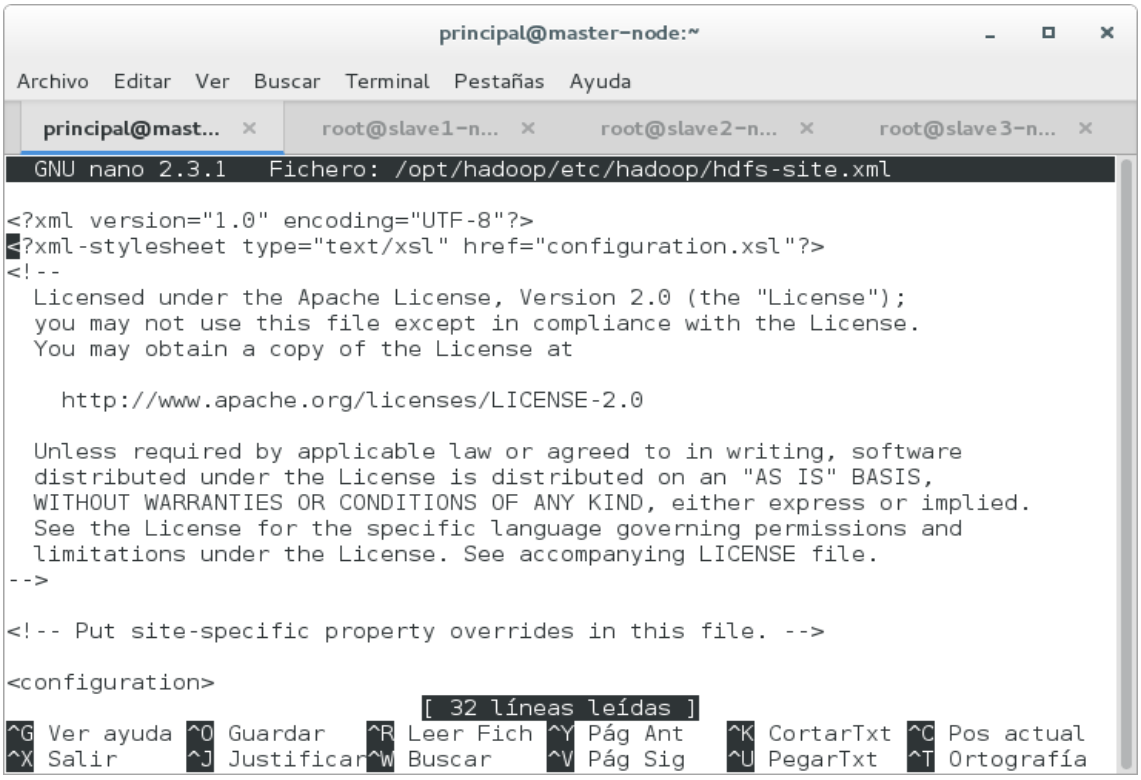


Figura 4.55: Archivo hdfs-site.xml *Nodo Master.*

- Ubicarse en la siguiente línea de código: `<configuration>`:

Figura 4.56: Archivo hdfs-site.xml *Nodo Master.*

- Se definen las principales propiedades de este archivo:

hdfs-site.xml

```
1. <configuration>
2. <property>
3. <name>dfs.replication</name>
4. <value>3</value>
5. </property>
6. <property>
7. <name>dfs.namenode.name.dir</name>
8. <value>file:/var/data/hadoop/hdfs/nn</value>
9. </property>
10.      <property>
11.      <name>dfs.namenode.checkpoint.dir</name>
12.      <value>file:/var/data/hadoop/hdfs/snn</value>
13.      </property>
14.      </configuration>
```

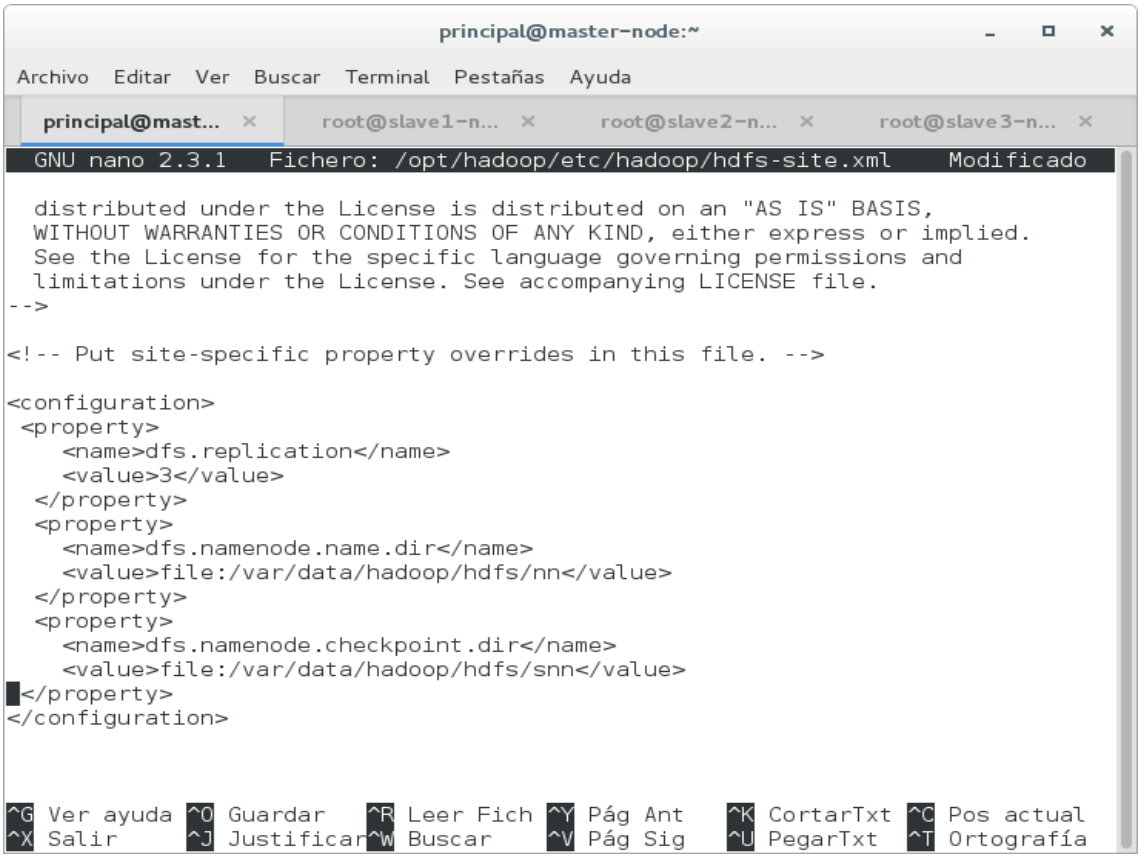


Figura 4.57: Agregación de líneas de código al archivo hdfs-site.xml *Nodo Master.*

Presionar Ctrl + X (para salir)

Como se puede visualizar en la parte superior existen tres líneas de código que se encuentran marcadas de color celeste: **<value>3</value>**, en esta línea de código se define el número de replicación de los nodos, es decir, el número de Nodos Esclavo que van a formar parte del clúster de Hadoop, en este caso se utilizarán tres Nodos Esclavo, ejecutados cada uno por separado en tres máquinas; **<value>file:/var/data/hadoop/hdfs/nn</value>**, en esta línea de código se establece el directorio correspondiente al Name Node y en esta línea de código, **<value>file:/var/data/hadoop/hdfs/snn</value>**, se establece el directorio correspondiente al Secondary Name Node.

Configuración del archivo *hdfs-site.xml* en los Nodos Esclavo:

- Se abre el *archivo hdfs-site.xml* digitando lo siguiente:

```
#nano /opt/hadoop/etc/hadoop/hdfs-site.xml
```

```
[root@slave1-node ~]# nano /opt/hadoop/etc/hadoop/hdfs-site.xml
```

Figura 4.58: Abrir archivo hdfs-site.xml *Nodos Esclavo.*

- Se abre el siguiente fichero correspondiente al *hdfs-site.xml* y ubicar el cursor en la línea de código: `<configuration>`:

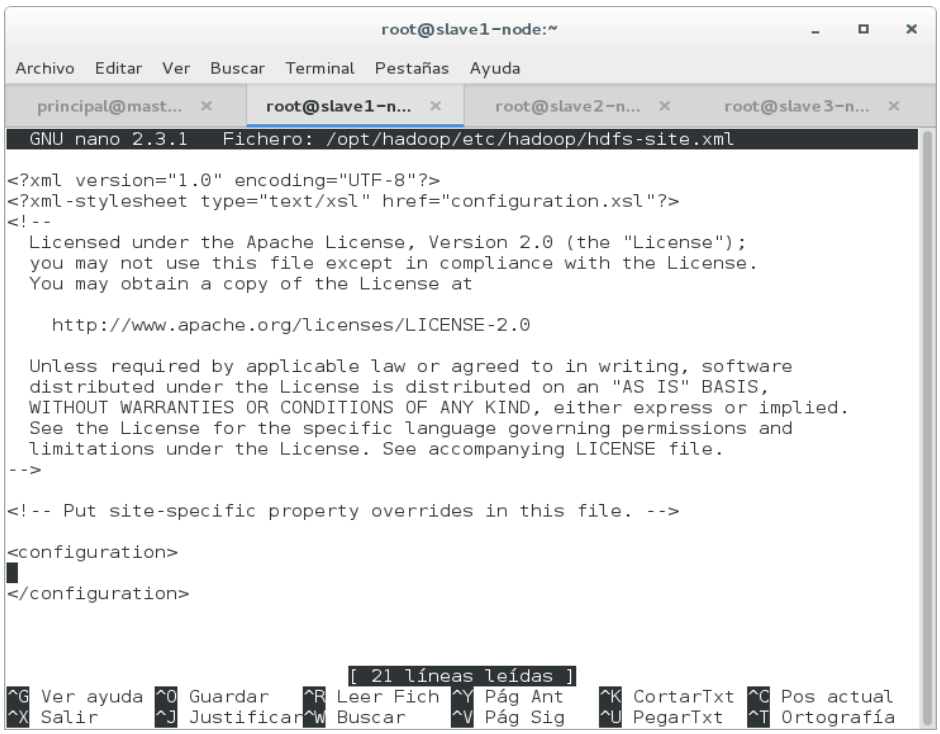


Figura 4.59: Archivo hdfs-site.xml *Nodos Esclavo.*

- Se definen las principales propiedades de este archivo que varían en cuanto a las propiedades del *hdfs-site.xml* del Nodo Master:

hdfs-site.xml

```
1. <configuration>
2. <property>
3. <name>dfs.replication</name>
4. <value>3</value>
5. </property>
6. <property>
7. <name>dfs.datanode.data.dir</name>
8. <value>file:/var/data/hadoop/hdfs/dn</value>
9. </property>
10.      </configuration>
```

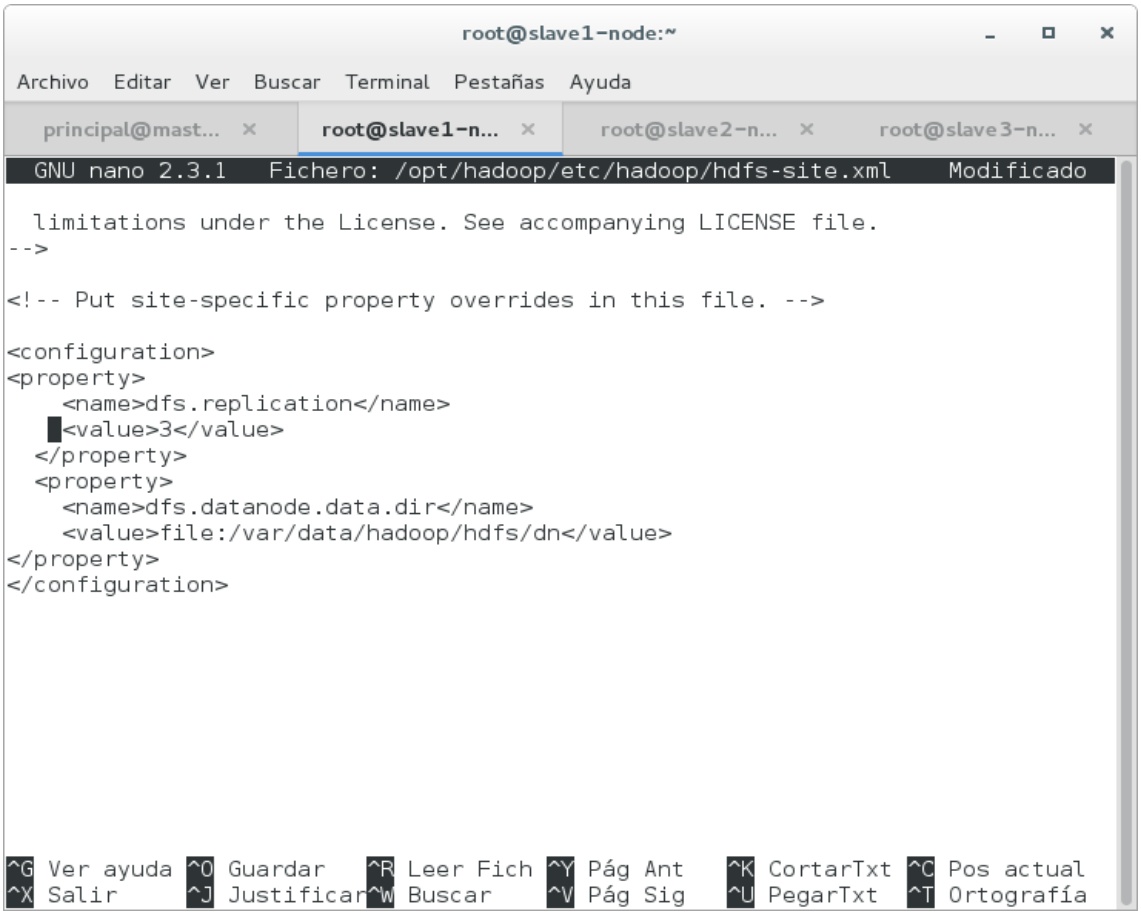


Figura 4.60: Agregación de líneas de código al archivo hdfs-site.xml *Nodos Esclavo.*

Presionar Ctrl + X (para salir)

Como se puede observar en la parte superior, existen dos líneas de código que se encuentran marcadas de color celeste para hacer referencia a las principales propiedades de este archivo: **<value>3</value>**, esta línea de código define el número de replicación de los nodos que al igual que en el archivo *hdfs-site.xml* del Nodo Master, el número de replicación sigue siendo 3 debido a que en este caso se trabajará con tres Nodos Esclavo y, **<value>file:/var/data/hadoop/hdfs/dn</value>**, esta línea define la ubicación del directorio correspondiente al Data Node.

Paso 8:

Configuración del archivo yarn-site.xml en el Nodo Master

- Se abre el archivo *yarn-site.xml* digitando lo siguiente:

```
#nano /opt/hadoop/etc/hadoop/yarn-site.xml
```

```
[root@master-node ~]# nano /opt/hadoop/etc/hadoop/yarn-site.xml
```

Figura 4.61: Abrir archivo yarn-site.xml *Nodo Master.*

- Se abre el archivo correspondiente al *yarn-site.xml* y se ubica el cursor en la línea de código `<configuration>`:

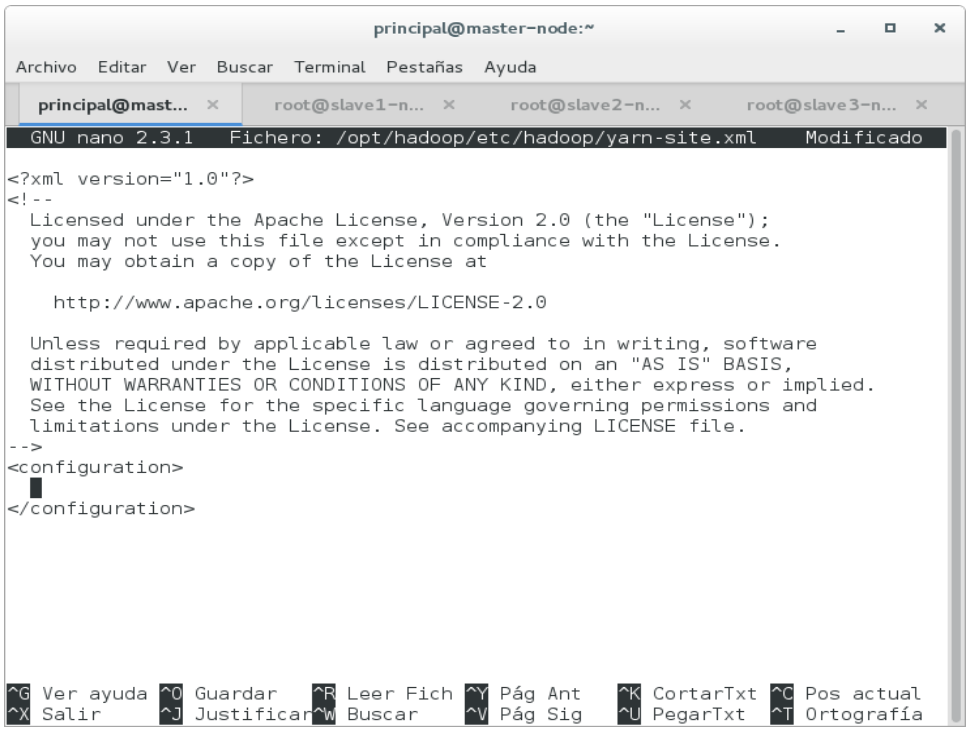


Figura 4.62: Archivo yarn-site.xml *Nodo Master.*

- Se definen las propiedades del archivo copiando las siguientes líneas de código:

yarn-site.xml

```
1. <configuration>
2. <property>
3. <name>yarn.resourcemanager.hostname</name>
4. <value>master-node.centos</value>
5. </property>
6. </configuration>
```

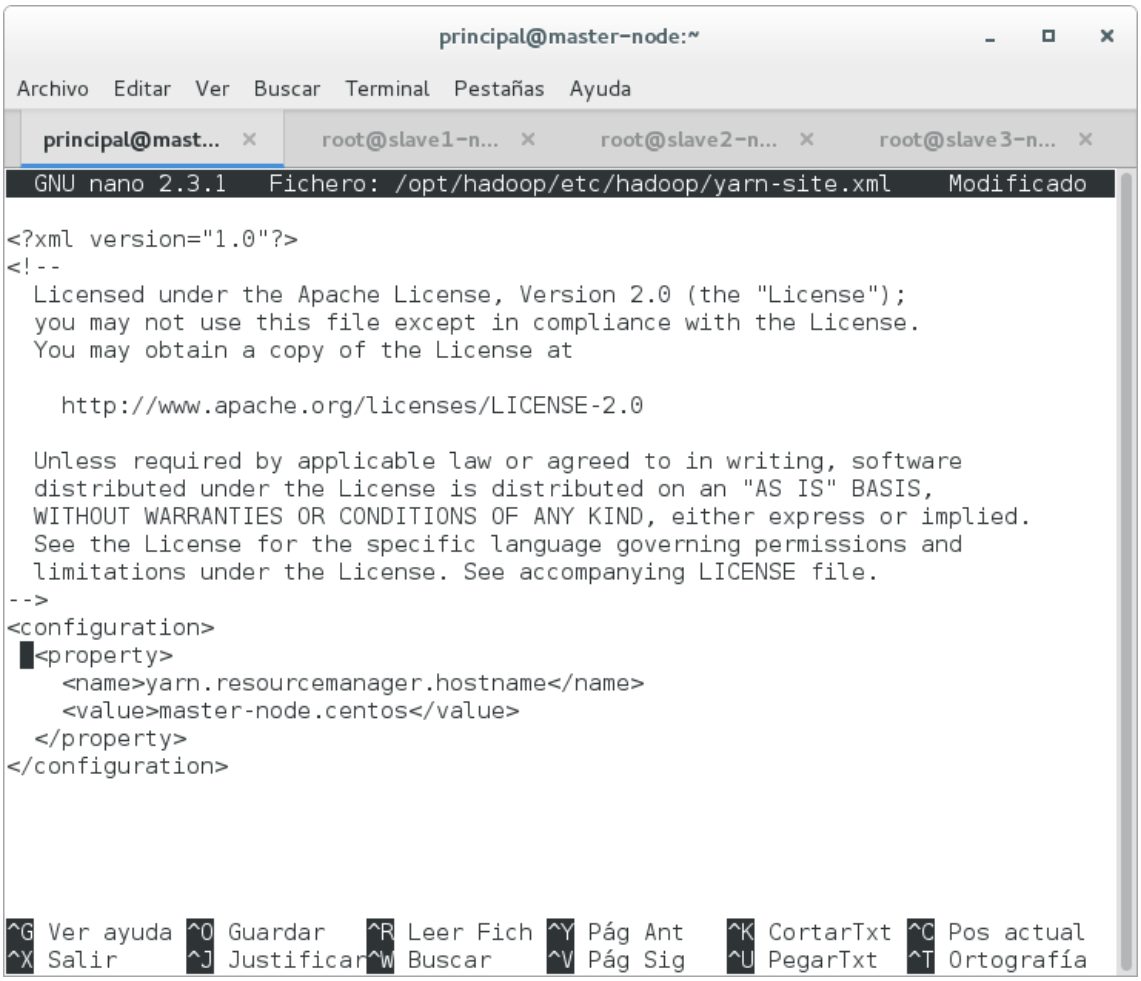


Figura 4.63: Agregación de líneas de código al archivo yarn-site.xml *Nodo Master.*

Presionar Ctrl + X (para salir)

Como se puede visualizar en la parte superior, la línea de código número 4 está marcada de color celeste: **<value>master-node.centos</value>**, ésta define el nodo en el cual se está ejecutando el Resource Manager. Como ya se mencionó en un apartado anterior en la presente configuración tanto el Name Node, el Secondary Name Node, el Job History Server

y el Resource Manager serán ejecutados en una sola máquina que trabajará como Nodo Master. En este caso la identificación completa de la máquina que ejecuta el Nodo Master (hostname) es: **master-node.centos**.

Configuración del archivo *yarn-site.xml* en los Nodos Esclavo

- Se abre el archivo *yarn-site.xml* digitando lo siguiente:

```
#nano /opt/hadoop/etc/hadoop/yarn-site.xml
```

```
[root@slave1-node ~]# nano /opt/hadoop/etc/hadoop/yarn-site.xml
```

Figura 4.64: Abrir archivo yarn-site.xml *Nodos Esclavo.*

- Se abre el archivo correspondiente al *yarn-site.xml* y se ubica el cursor en la línea de código `<configuration>`:

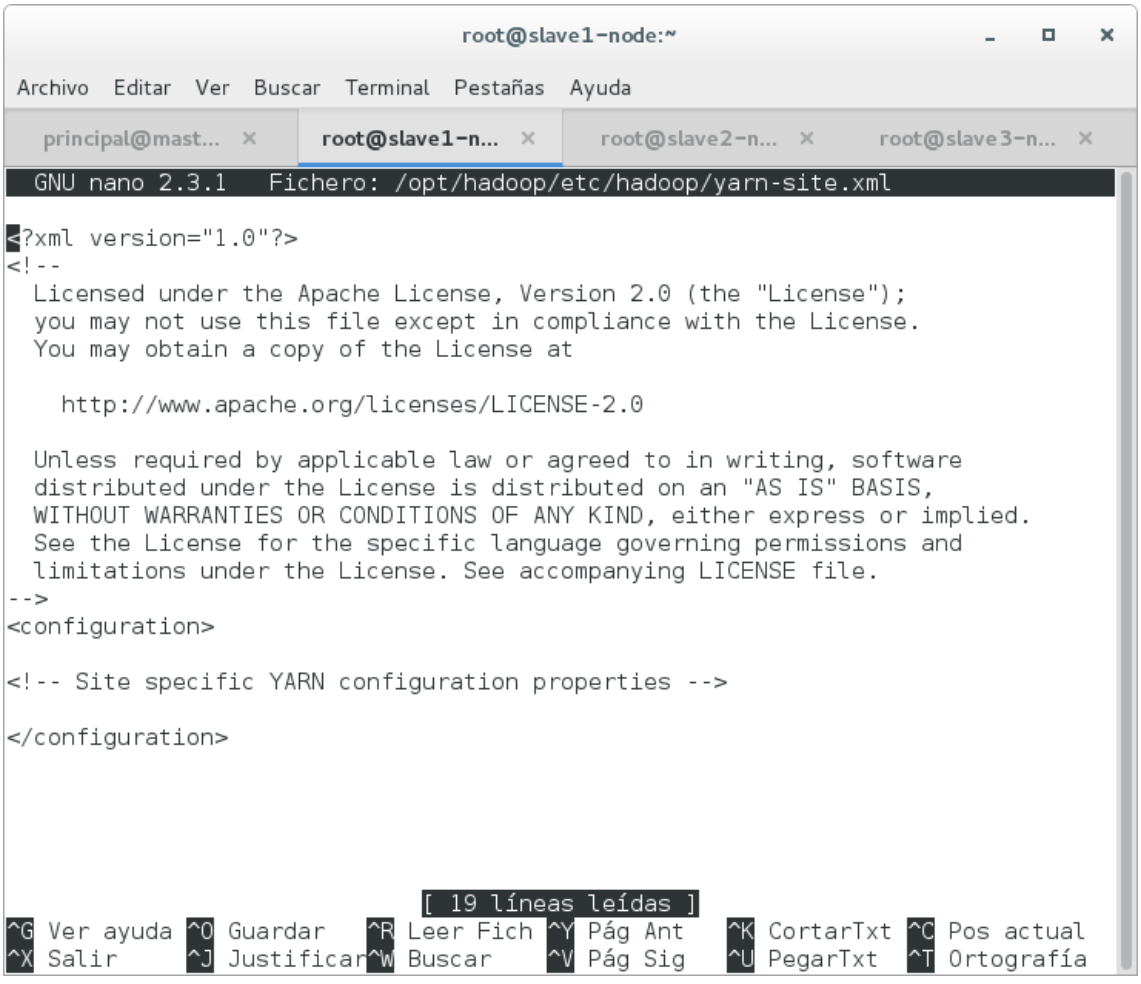


Figura 4.65: Archivo yarn-site.xml *Nodos Esclavo.*

- Se definen las propiedades del archivo copiando las siguientes líneas de código, debido a que el archivo *yarn-site.xml* de los Nodos Esclavo es diferente al correspondiente archivo del Nodo Master.

yarn-site.xml

```
1. <configuration>
2. <property>
3. <name>yarn.nodemanager.aux-services</name>
4. <value>mapreduce_shuffle</value>
5. </property>
6. <property>
7. <name>yarn.nodemanager.aux-services.mapreduce_shuffle.class</name>
8. <value>org.apache.hadoop.mapred.ShuffleHandler</value>
9. </property>
10.      <property>
11.      <name>yarn.nodemanager.log-dirs</name>
12.      <value>file:/var/hadoop/yarn/logs</value>
13.      </property>
14.      <property>
15.      <name>yarn.nodemanager.local-dirs</name>
16.      <value>file:/var/hadoop/yarn/local</value>
17.      </property>
18.      <property>
19.      <name>yarn.resourcemanager.hostname</name>
20.      <value>master-node.centos</value>
21.      </property>
22.      </configuration>
```

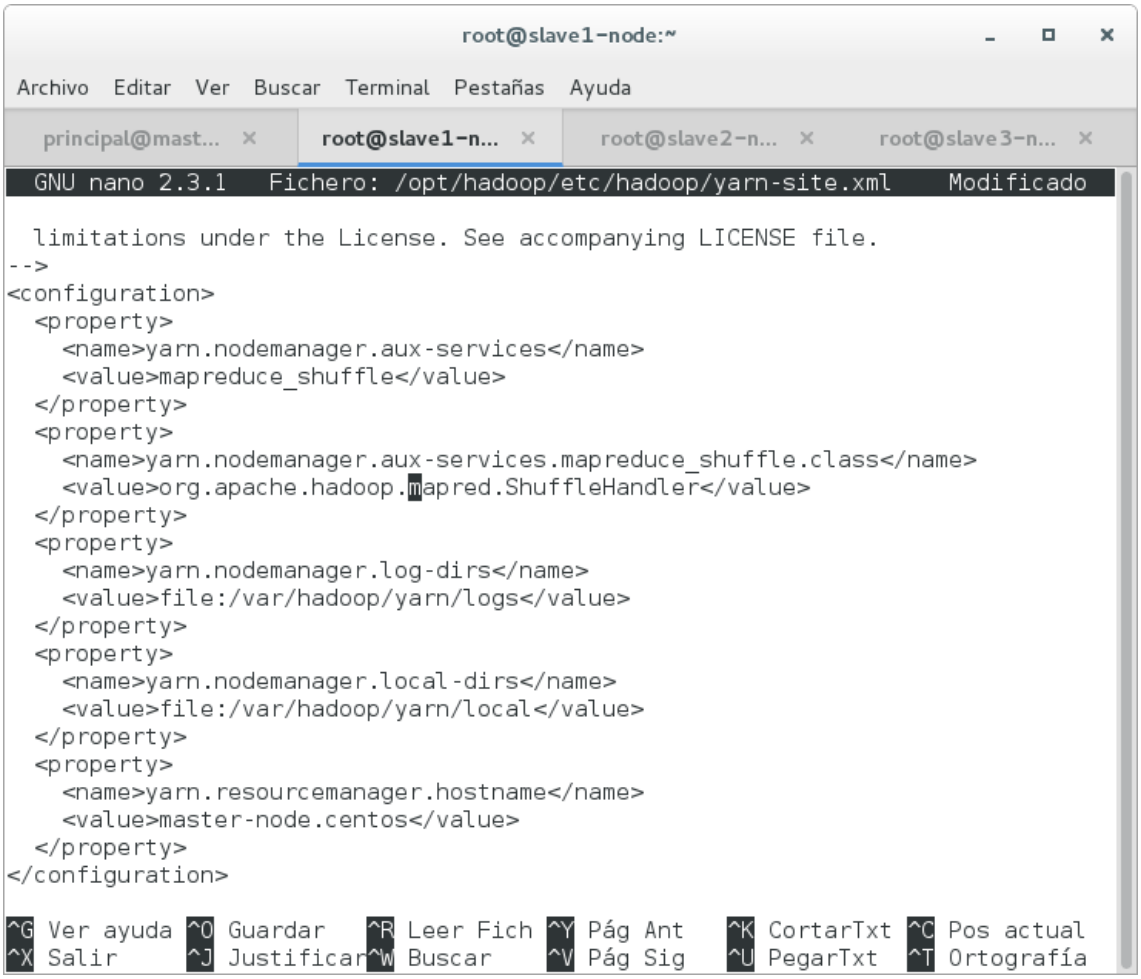


Figura 4.66: Agregación de líneas de código al archivo yarn-site.xml *Nodos Esclavo.*

Presionar Ctrl + X (para salir)

Como se puede visualizar las siguientes líneas de código se encuentran marcadas de color celeste:

```
12.<value>file:/var/hadoop/yarn/logs</value>
```

```
16.<value>file:/var/hadoop/yarn/local</value>
```

Éstas definen la ubicación de los directorios que van a ser utilizados por el Node Manager.

Por otro lado, en esta línea de código: `<value>master-node.centos</value>` se define el nodo en el cual se ejecuta el componente Resource Manager, el mismo que se encuentra en el Nodo Master, por lo tanto, se debe colocar el hostname de la máquina correspondiente al Nodo Master: `master-node.centos`.

PASO 9:

Edición del archivo *eslaves* en el Nodo Master

- Se abre el archivo *slaves* digitando lo siguiente:

```
#nano /opt/hadoop/etc/hadoop/slaves
```

```
[root@master-node ~]# nano /opt/hadoop/etc/hadoop/slaves
```

Figura 4.67: Abrir fichero slaves *para agregar los Nodos Esclavo.*

- Agregar los nodos que se van a ejecutar como Esclavos:

```
slave1-node.centos
slave2-node.centos
slave3-node.centos
```

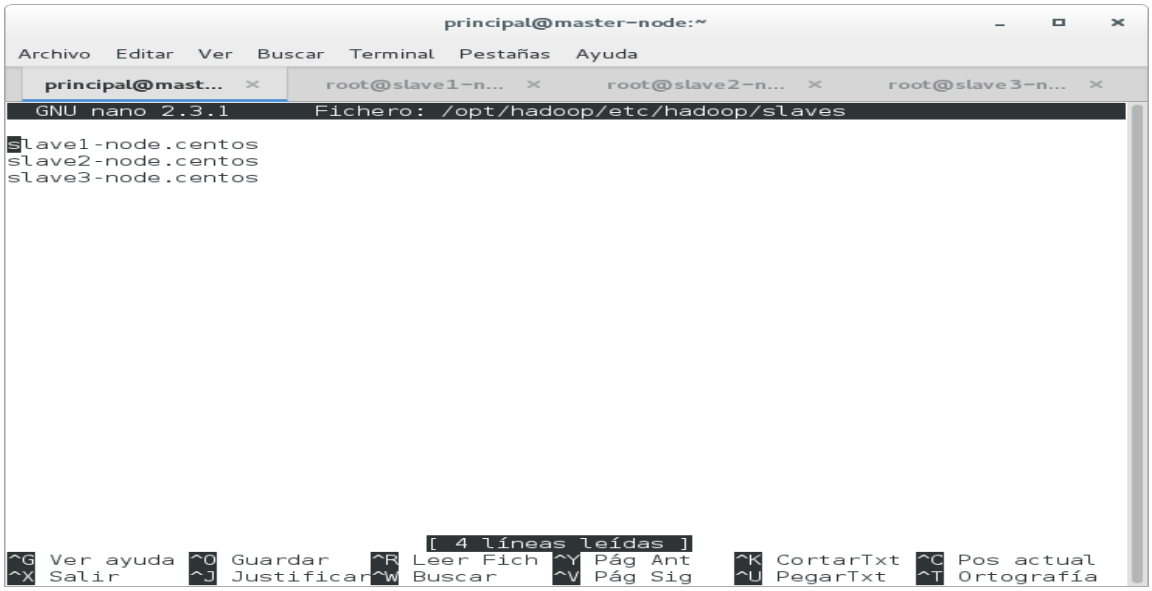


Figura 4.68: Agregación de los Nodos Esclavo al fichero slaves.

Presionar Ctrl + X (para salir)

En este fichero se deben agregar las identificaciones completas de las máquinas (hostnames) en las cuales se están ejecutando los Nodos Eclavo, este caso se tienen tres máquinas que funcionan como Nodos Esclavo y sus identificaciones con sus correspondientes dominios son: *slave1-node.centos*, *slave2-node.centos* y *slave3-node.centos*.

PASO 10:

Parar el Firewall del Sistema en el Nodo Master

- Digitar el siguiente comando para interrumpir el Firewall:

```
#systemctl stop firewalld
```

```
[root@master-node ~]# systemctl stop firewalld
```

Figura 4.69: Detener el funcionamiento del Firewall del sistema en el Nodo Master.

Este paso es importante realizarlo ya que si el firewall se encuentra encendido podrían ocurrir problemas tanto en la comunicación como en el intercambio de información entre los diferentes Nodos que conforman el clúster de Hadoop.

Desactivar el IPv6 del Sistema en el Nodo Master

- Abrir el archivo de configuración digitando lo siguiente:

```
#nano /etc/sysctl.conf
```

```
[root@master-node ~]# nano /etc/sysctl.conf
```

Figura 4.70: Abrir fichero sysctl.conf *en el Nodo Master.*

- Agregar las siguientes líneas de código que desactivarán IPv6:

```
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
```

Figura 4.71: Desactivación del IPv6 en el Nodo Master.

Presionar Ctrl + X (para salir)

Es importante realizar esta configuración debido a que Hadoop no es compatible con IPv6.

Parar el Firewall del Sistema en los Nodos Esclavo

- Digitar el siguiente comando para interrumpir el Firewall:

```
#systemctl stop firewalld
```

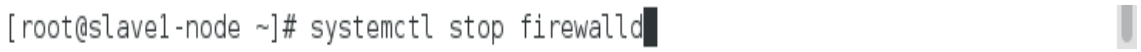


Figura 4.72: Detener el funcionamiento del Firewall del sistema en los Nodos Esclavo.

Desactivar el IPv6 del Sistema en los Nodos Esclavo

- Abrir el archivo de configuración digitando lo siguiente:

```
#nano /etc/sysctl.conf
```

Figura 4.73: Abrir fichero sysctl.conf *en los Nodos Esclavo.*

- Agregar las siguientes líneas de código que desactivarán IPv6:

```
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
```

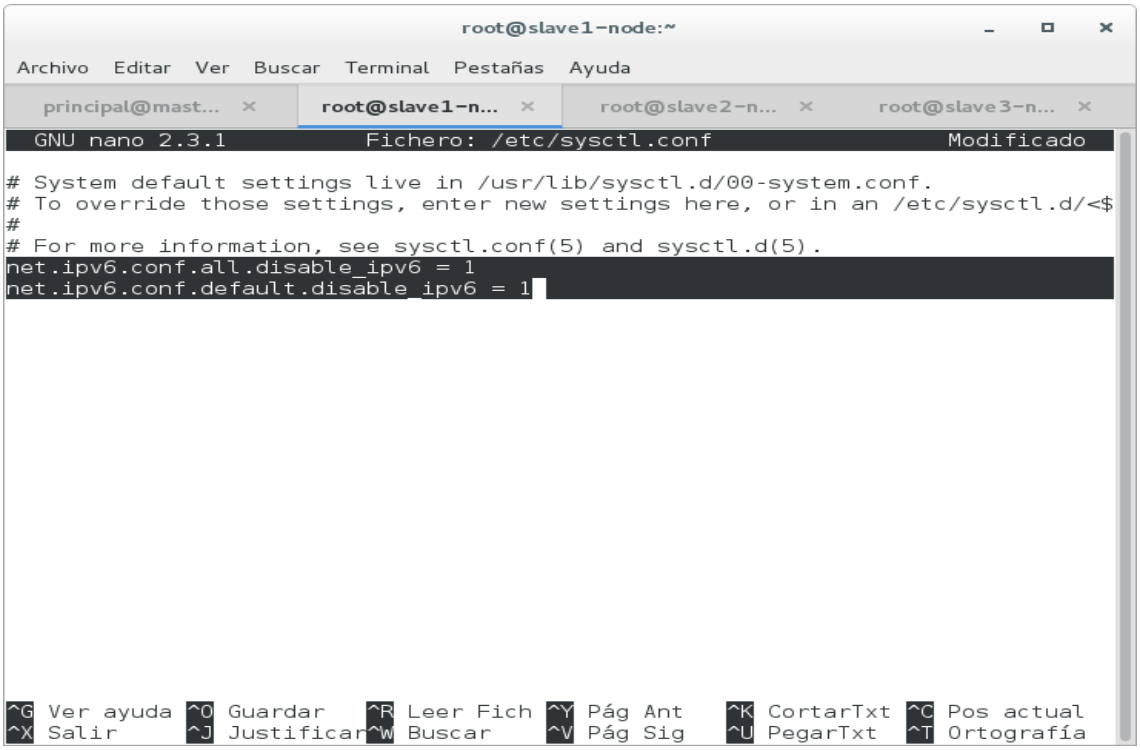


Figura 4.74: Desactivación del IPv6 en los Nodos Esclavo.

Presionar Ctrl + X (para salir)

PASO 11:

Dar Formato al Name Node

Este paso debe ser ejecutado únicamente en el Nodo Master.

- Iniciar sesión como usuario HDFS digitando lo siguiente:

```
#su hdfs
```

```
[root@master-node ~]# su hdfs
```

Figura 4.75: Ingreso como usuario hdfs.

- Buscar el Directorio *bin* de Hadoop:

```
$cd /opt/hadoop/bin/
```

```
[hdfs@master-node root]$ cd /opt/hadoop/bin/
```

*Figura 4.76: Ingreso a la carpeta /*bin *del usuario hdfs.*

- Ejecutar el siguiente comando para dar formato al sistema de Archivos:

```
$./hdfs namenode -format
```

```
[hdfs@master-node bin]$ ./hdfs namenode -format
```

Figura 4.77: Formato al Name Node.

- Se observa el siguiente resultado:

```
hdfs@master-node:/opt/hadoop/bin
Archivo  Editar  Ver  Buscar  Terminal  Pestañas  Ayuda
hdfs@master-node:/opt/hadoop/bin   root@slave1-node:~   root@slave2-node:~   root@slave3-node:~
16/06/08 13:00:36 INFO blockmanagement.BlockManager: maxNumBlocksToLog          = 1000
16/06/08 13:00:36 INFO namenode.FSNamesystem: fsOwner             = hdfs (auth:SIMPLE)
16/06/08 13:00:36 INFO namenode.FSNamesystem: supergroup          = supergroup
16/06/08 13:00:36 INFO namenode.FSNamesystem: isPermissionEnabled = true
16/06/08 13:00:36 INFO namenode.FSNamesystem: HA Enabled: false
16/06/08 13:00:36 INFO namenode.FSNamesystem: Append Enabled: true
16/06/08 13:00:37 INFO util.GSet: Computing capacity for map INodeMap
16/06/08 13:00:37 INFO util.GSet: VM type       = 64-bit
16/06/08 13:00:37 INFO util.GSet: 1.0% max memory 889 MB = 8.9 MB
16/06/08 13:00:37 INFO util.GSet: capacity      = 2^20 = 1048576 entries
16/06/08 13:00:37 INFO namenode.FSDirectory: ACLs enabled? false
16/06/08 13:00:37 INFO namenode.FSDirectory: XAttrs enabled? true
16/06/08 13:00:37 INFO namenode.FSDirectory: Maximum size of an xattr: 16384
16/06/08 13:00:37 INFO namenode.NameNode: Caching file names occuring more than 10 times
16/06/08 13:00:37 INFO util.GSet: Computing capacity for map cachedBlocks
16/06/08 13:00:37 INFO util.GSet: VM type       = 64-bit
16/06/08 13:00:37 INFO util.GSet: 0.25% max memory 889 MB = 2.2 MB
16/06/08 13:00:37 INFO util.GSet: capacity      = 2^18 = 262144 entries
16/06/08 13:00:37 INFO namenode.FSNamesystem: dfs.namenode.safemode.threshold-pct = 0.9990000128746033
16/06/08 13:00:37 INFO namenode.FSNamesystem: dfs.namenode.safemode.min.datanodes = 0
16/06/08 13:00:37 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension     = 30000
16/06/08 13:00:37 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
16/06/08 13:00:37 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
16/06/08 13:00:37 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
16/06/08 13:00:37 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
16/06/08 13:00:37 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
16/06/08 13:00:37 INFO util.GSet: Computing capacity for map NameNodeRetryCache
16/06/08 13:00:37 INFO util.GSet: VM type       = 64-bit
16/06/08 13:00:37 INFO util.GSet: 0.029999999329447746% max memory 889 MB = 273.1 KB
16/06/08 13:00:37 INFO util.GSet: capacity      = 2^15 = 32768 entries
Re-format filesystem in Storage Directory /var/data/hadoop/hdfs/nn ? (Y or N) Y
16/06/08 13:00:42 INFO namenode.FSImage: Allocated new BlockPoolId: BP-206187773-192.168.1.100-1465408842136
16/06/08 13:00:42 INFO common.Storage: Storage directory /var/data/hadoop/hdfs/nn has been successfully formatted.
16/06/08 13:00:42 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
16/06/08 13:00:42 INFO util.ExitUtil: Exiting with status 0
16/06/08 13:00:42 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at master-node.centos/192.168.1.100
************************************************************/
[hdfs@master-node bin]$
```

Figura 4.78: Resultado de dar formato al Name Node.

PASO 12:

Iniciar Componentes o Demonios Correspondientes al Nodo Master

Realizar los siguientes pasos únicamente en el Nodo Master.

- Ir al directorio *sbin* de Hadoop digitando lo siguiente:

```
$cd /opt/hadoop/sbin/
```

```
[hdfs@master-node bin]$ cd /opt/hadoop/sbin/
```

*Figura 4.79: Ir al directorio /*sbin *del usuario hdfs.*

Iniciar el Demonio Name Node

- Ejecutar el comando de inicio del Name Node:

```
$./hadoop-daemon.sh start namenode
```

```
[hdfs@master-node sbin]$ ./hadoop-daemon.sh start namenode
starting namenode, logging to /opt/hadoop/logs/hadoop-hdfs-namenode-master-node.
centos.out
```

Figura 4.80: Inicio del demonio Name Node.

Iniciar el Demonio Secondary Name Node

- Ejecutar el comando de inicio del Secondary Name Node:

```
$./hadoop-daemon.sh start secondarynamenode
```

```
[hdfs@master-node sbin]$ ./hadoop-daemon.sh start secondarynamenode
starting secondarynamenode, logging to /opt/hadoop/logs/hadoop-hdfs-secondarynam
enode-master-node.centos.out
```

Figura 4.81: Inicio del demonio Secondary Name Node.

PASO 13:

Iniciar los Data Node en los Nodos Esclavo

Realizar estos pasos en todos los Nodos Esclavo.

- Iniciar sesión con el usuario hdfs digitando lo siguiente:

```
#su hdfs
```

```
[root@slave1-node ~]# su hdfs
```

Figura 4.82: Usuario hdfs Nodos Esclavo.

- Cambiar al directorio *sbin* de Hadoop:

```
$cd /opt/hadoop/sbin
```

```
[hdfs@slave1-node root]$ cd /opt/hadoop/sbin
```

Figura 4.83: Fichero /sbin *usuario hdfs Nodos Esclavos.*

❖ Ejecutar el siguiente comando para iniciar el demonio Data Node:

```
$./hadoop-daemon.sh start datanode
```

```
[hdfs@slave1-node sbin]$ ./hadoop-daemon.sh start datanode
starting datanode, logging to /opt/hadoop/logs/hadoop-hdfs-datanode-slave1-node.
centos.out
```

Figura 4.84: Inicio del demonio Data Node.

PASO 14:

Iniciar YARN en el Nodo Master:

❖ Salir del directorio *sbin* del usuario hdfs e iniciar sesión con el usuario Yarn:

- Para salir del directorio *sbin* del usuario hdfs*:*

```
$exit
```

- Para iniciar sesión con el usuario Yarn:

```
#su yarn
```

```
[hdfs@master-node sbin]$ exit
exit
[root@master-node ~]# su yarn
```

Figura 4.85: Usuario yarn Nodo Master.

❖ Buscar el directorio sbin del usuario Yarn:

```
$cd /opt/hadoop/sbin/
```

```
[yarn@master-node root]$ cd /opt/hadoop/sbin/
```

*Figura 4.86: Fichero /*sbin *usuario yarn Nodo Master.*

❖ Ejecutar el comando para iniciar el demonio Resource Manager:

```
$./yarn-daemon.sh start resourcemanager
```

```
[yarn@master-node sbin]$ ./yarn-daemon.sh start resourcemanager
starting resourcemanager, logging to /opt/hadoop/logs/yarn-yarn-resourcemanager-
master-node.centos.out
```

Figura 4.87: Inicio del demonio Resource Manager.

Iniciar YARN en los Nodos Esclavo

Realizar estos pasos en todos los Nodos Esclavo.

- ❖ Salir del directorio *sbin* del usuario hdfs e iniciar sesión con el usuario Yarn:
- Para salir del directorio *sbin* del usuario hdfs*:*

```
$exit
```

- Para iniciar sesión con el usuario Yarn:

```
#su yarn
```

```
[hdfs@slave1-node sbin]$ exit
exit
[root@slave1-node ~]# su yarn
```

Figura 4.88: Usuario yarn Nodos Esclavo.

- ❖ Buscar el directorio *sbin* del usuario Yarn:

```
$cd /opt/hadoop/sbin/
```

```
[yarn@slave1-node root]$ cd /opt/hadoop/sbin/
```

*Figura 4.89: Directorio /*sbin *del usuario yarn en los Nodos Esclavo.*

- ❖ Ejecutar el comando para iniciar el demonio Node Manager:

```
$./yarn-daemon.sh start nodemanager
```

```
[yarn@slave1-node sbin]$ ./yarn-daemon.sh start nodemanager
starting nodemanager, logging to /opt/hadoop/logs/yarn-yarn-nodemanager-slave1-n
ode.centos.out
```

Figura 4.90: Inicio del demonio Node Manager.

Verificar que los Componentes Data Node y Node Manager se estén ejecutando en Todos los Nodos Esclavo:

Realizar lo siguiente en todos los nodos esclavo.

- ❖ Salir del usuario Yarn:

```
$exit
```

```
[yarn@slave1-node sbin]$ exit
exit
```

Figura 4.91: Salir del usuario yarn Nodos Esclavo.

- Ejecutar el comando jps para comprobar que se estén ejecutando los componentes Data Node y Node Manager:

```
#jps
```

```
[root@slave1-node ~]# jps
3584 NodeManager
3299 DataNode
3702 Jps
```

Figura 4.92: Verificación de los demonios activos en los Nodos Esclavo.

Como se puede observar los componentes Node Manager y Data Node se están ejecutando correctamente en los Nodos Esclavo, si estos competentes no se visualizan como lo muestra la imagen anterior, es probable que la configuración haya fallado en algún punto.

PASO 15:

Crear Directorios en el Nodo Master

- Salir del usuario Yarn:

```
$exit
```

```
[yarn@master-node sbin]$ exit
exit
```

Figura 4.93: Salir del usuario yarn en el Nodo Master.

- Iniciar sesión con el usuario hdfs:

```
#su hdfs
```

```
[root@master-node ~]# su hdfs
```

Figura 4.94: Ingreso como usuario hdfs en Nodo Master.

- Entrar al directorio *bin*:

```
$cd /opt/hadoop/bin/
```

```
[hdfs@master-node root]$ cd /opt/hadoop/bin/
```

Figura 4.95: Directorio /bin *del usuario hdfs en Nodo Master.*

- Crear los directorios *user* y *temp* en el sistema de archivos de Hadoop:

```
$./hdfs dfs –mkdir –p /user
$./hdfs dfs –chmod 777 /user
$./hdfs dfs –mkdir –p /tmp
$./hdfs dfs –chmod 777 /tmp
```

```
[hdfs@master-node bin]$ ./hdfs dfs -mkdir -p /user
[hdfs@master-node bin]$ ./hdfs dfs -chmod 777 /user
[hdfs@master-node bin]$ ./hdfs dfs -mkdir -p /tmp
[hdfs@master-node bin]$ ./hdfs dfs -chmod 777 /tmp
```

Figura 4.96: Creación de directorios user *y* temp *en Nodo Master.*

- Verificar que se hayan creado correctamente los directorios:

```
$./hdfs dfs –ls /
```

```
[hdfs@master-node bin]$ ./hdfs dfs -ls /
16/06/24 13:22:53 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your pl
atform... using builtin-java classes where applicable
Found 2 items
drwxrwxrwx   - hdfs supergroup          0 2016-06-24 13:22 /tmp
drwxrwxrwx   - hdfs supergroup          0 2016-06-24 13:22 /user
```

Figura 4.97: Verificación de los directorios creados.

La advertencia que se muestra en la imagen anterior aparece cuando no se utiliza el java que viene por defecto en los sistemas operativos LINUX, sino cuando se utiliza el java instalado desde la página oficial tal como se realizó en el primer paso de esta práctica.

PASO 16:

Iniciar el demonio Job History Server en el Nodo Master

- Salir del usuario HDFS:

```
$exit
```

```
[hdfs@master-node bin]$ exit
```

Figura 4.98: Salir del usuario hdfs en el Nodo Master.

- Iniciar sesión con el usuario mapred:

```
#su mapred
```

```
[root@master-node ~]# su mapred
```

Figura 4.99: Sesión como usuario mapred en Nodo Master.

- Entrar al directorio *sbin*:

```
$cd /opt/hadoop/sbin/
```

```
[mapred@master-node root]$ cd /opt/hadoop/sbin/
```

Figura 4.100: Directorio /sbin del usuario mapred en Nodo Master.

- Ejecutar el comando para iniciar el componente Job History Server:

```
$./mr-jobhistory-daemon.sh start historyserver
```

```
[mapred@master-node sbin]$ ./mr-jobhistory-daemon.sh start historyserver
chown: cambiando el propietario de «/opt/hadoop/logs»: Operación no permitida
starting historyserver, logging to /opt/hadoop/logs/mapred-mapred-historyserver-master-node
.centos.out
```

Figura 4.101: Inicio del demonio Job Histoy Server.

PASO 18:

Ejecución de un Ejemplo de Hadoop que viene por defecto

A continuación, se ejecutará un ejemplo que viene por defecto en Hadoop para comprobar el correcto funcionamiento tanto del Nodo Master como de los Nodos Esclavo que se encuentran conformando el clúster de Hadoop.

*Seguir con la ejecución en el Nodo Master.

- Cambiar al directorio *bin* de Hadoop:

```
$cd /opt/hadoop/bin/
```

```
[mapred@master-node sbin]$ cd /opt/hadoop/bin/
```

*Figura 4.102: Directorio /*bin *del usuario mapred en Nodo Master.*

❖ Ejecutar el ejemplo que calcula “pi”:

Se eligió la ejecución del “pi”, para demostrar que Hadoop es útil para diferentes casos, como por ejemplo, para el cálculo de problemas matemáticos. Puede parecer irrelevante el calcular el número “pi”, pero debido a que este es un número infinito, resulta interesante ver como los sistemas informáticos pueden calcular pequeños trozos de este número casi en poco tiempo, lo que demuestra que se puede utilizar esta herramienta en combinación con algoritmos para cálculos criptográficos, estadísticos, minería de datos, física, etc. A continuación se calcula “pi” 10 10, lo que quiere decir que se usan 10 asignaciones con 10 ejemplo de cada una para calcular su valor.

```
$./yarn jar /opt/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-
examples-2.7.2.jar pi 10 10
```

```
[mapred@master-node bin]$ ./yarn jar /opt/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-ex
amples-2.7.2.jar pi 10 10
```

Figura 4.103: Ejecución del ejemplo pi *que viene por defecto en Hadoop.*

Se pueden visualizar los siguientes resultados:

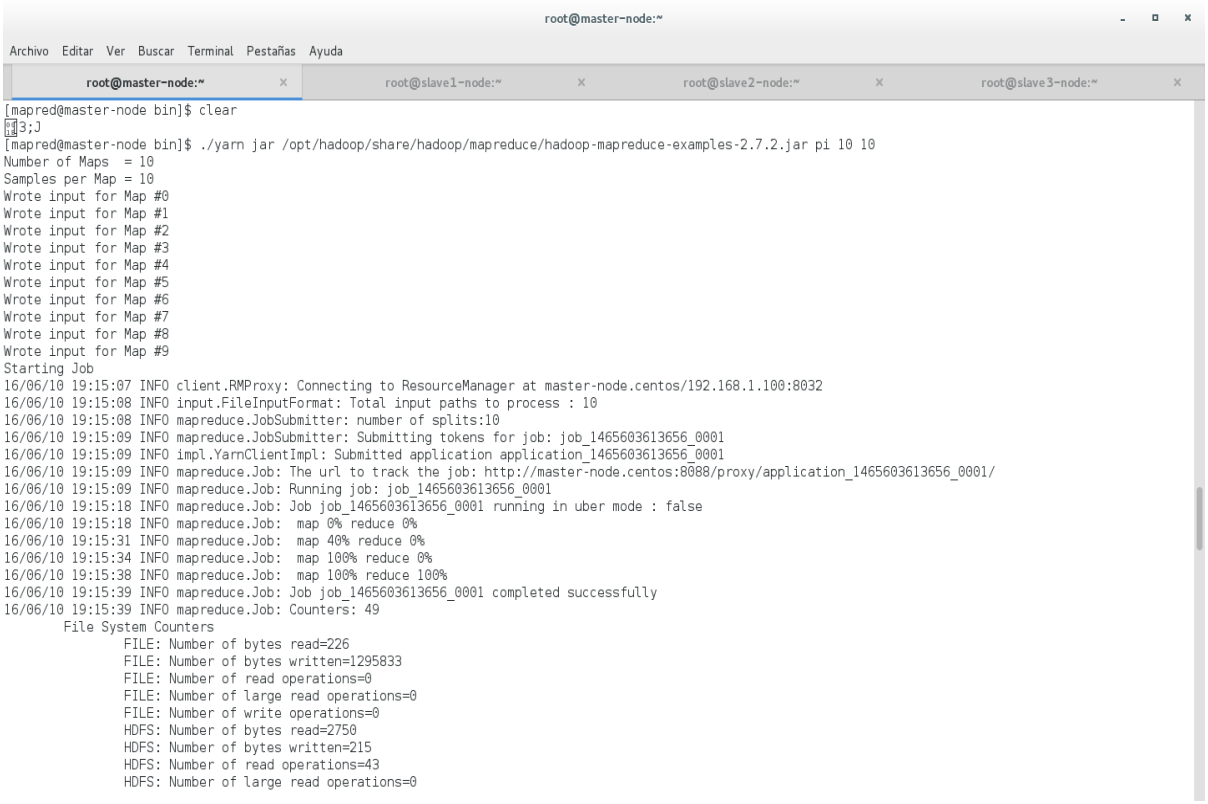


Figura 4.104: Resultado de la ejecución del ejemplo pi *en Hadoop.*

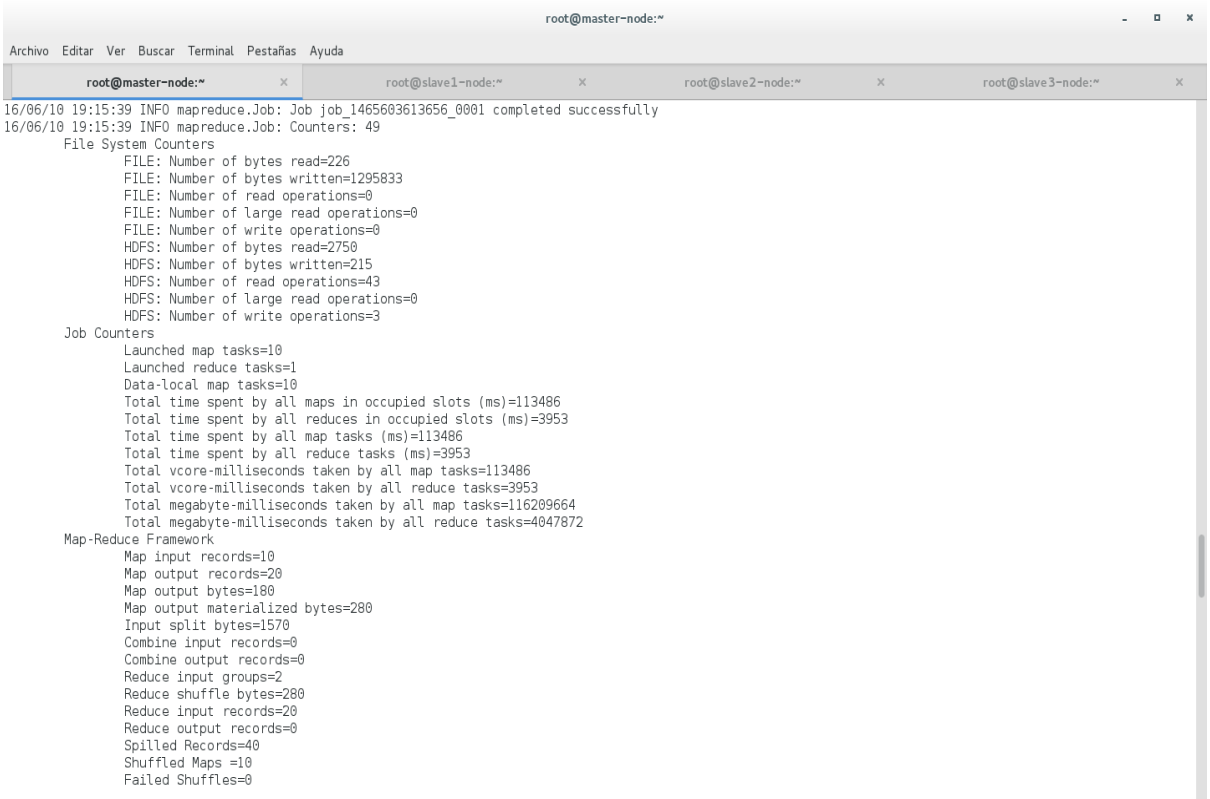


Figura 4.105: Resultado de la ejecución del ejemplo pi *en Hadoop.*

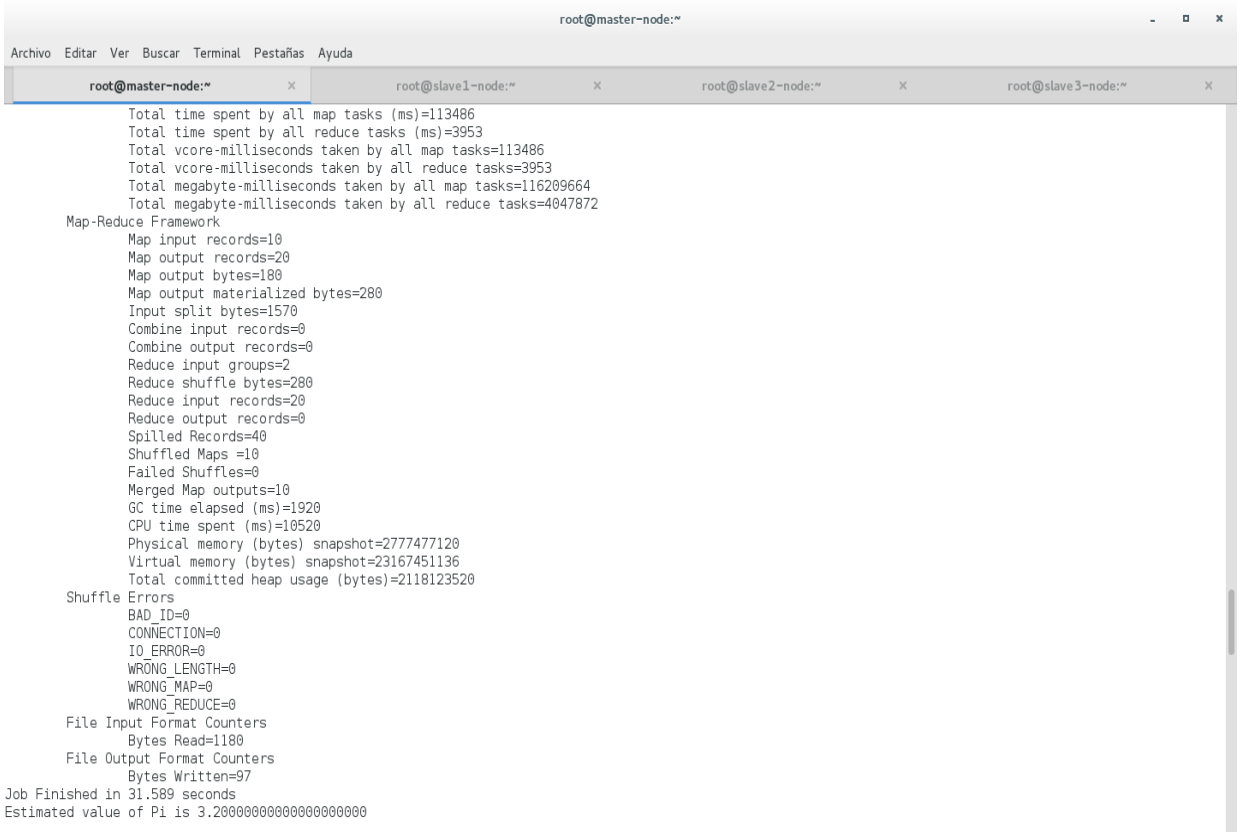


Figura 4.106: Resultado de la ejecución del ejemplo pi *en Hadoop.*

Como se puede observar la ejecución de este ejemplo tuvo una duración de 31.589 segundos y se ejecutó correctamente en los tres nodos esclavo. Si se trabajara con un solo nodo esclavo el tiempo sería más extenso y si se trabajara con más de tres nodos esclavo el tiempo iría disminuyendo conforme existan más nodos de replicación.

- ❖ Ahora, verificar que los componentes del Nodo Master estén funcionando correctamente.

Digitar los siguientes comandos:

- Para salir del usuario mapred:

```
$ exit
```

- Para comprobar el estado de los componentes:

```
# jps
```

```
[root@master-node ~]# jps
9348 SecondaryNameNode
10404 JobHistoryServer
9254 NameNode
12215 Jps
9498 ResourceManager
```

Figura 4.107: Comprobación de los componentes que se están ejecutando.

Como se puede observar en la captura anterior, todos los componentes correspondientes al Nodo Master están encendidos.

- Ahora, es necesario ir al explorador para observar el funcionamiento tanto del Nodo Master como de los Nodos Esclavo.

Para poder observar los resultados de todos los Nodos en conjunto en el explorador, es necesario colocar en la barra de direcciones el nombre que se utilizó para identificar a la máquina correspondiente al Nodo Master, mismo que fue situado en el archivo ***core-site.xml*** del Nodo Master, para este caso la identificación de la máquina es ***master-node.centos***, entonces se debe digitar en la barra de direcciones del explorador ese nombre seguido del puerto que tiene Hadoop por defecto que es el 8088, este puerto permite obtener la información sobre el clúster y todas la aplicaciones que se estén ejecutando, de esta manera se observarán los resultados de todos los Nodos y procesos que se están ejecutando en Hadoop, tal como se muestra en las siguientes imágenes.

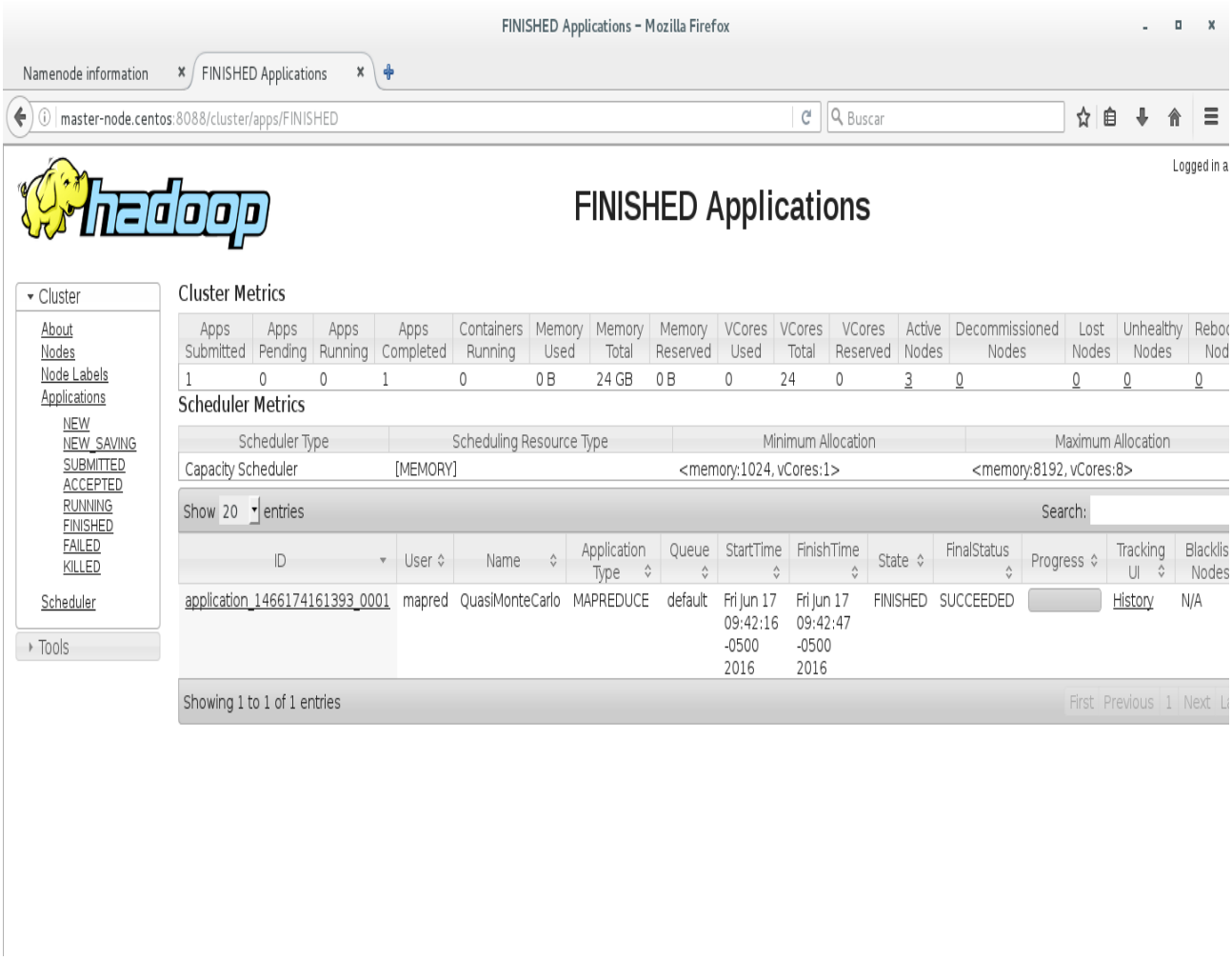


Figura 4.108: Vista en el explorador de los resultados del proceso ejecutado en Hadoop.

En la imagen anterior se puede observar que al seleccionar la opción “Applications” en el menú que se encuentra en la parte izquierda de la pantalla, se despliega la información de la aplicación que fue ejecutada en los nodos de Hadoop, el estado de los procesos y los recursos consumidos al ejecutar las aplicaciones. En este caso, se puede observar que la aplicación que fue ejecutada fue la de “***QuasiMonteCarlo”***, la misma que contenía el ejemplo ***pi*** que fue utilizado para esta demostración.

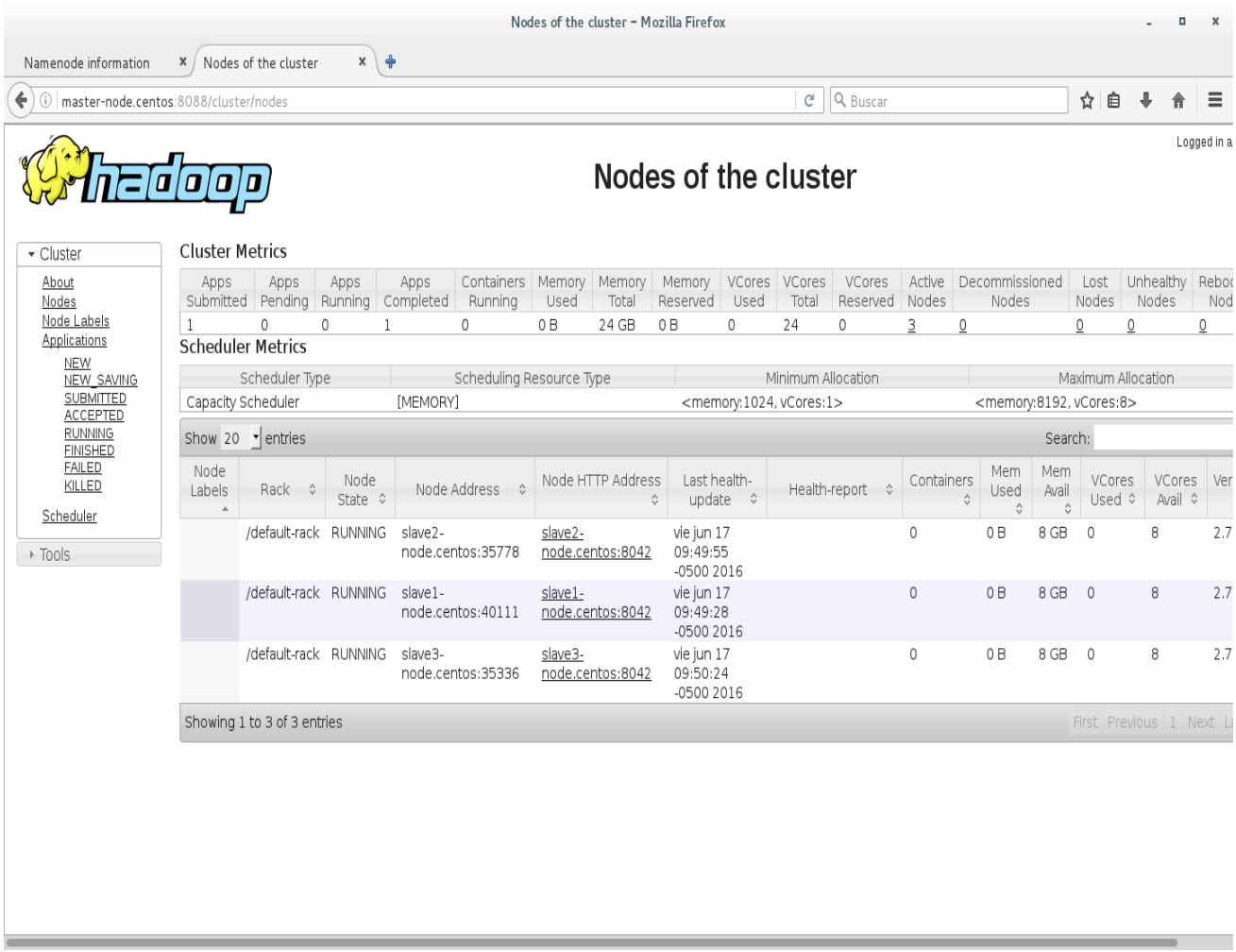


Figura 4.109: Vista en el explorador de los resultados del proceso ejecutado en Hadoop.

En la imagen anterior se puede observar que al seleccionar la opción “Nodes”, en el mismo menú, se despliega la información de todos los Nodos Esclavo que participaron en la ejecución de una aplicación de Hadoop, así como la fecha y la hora de ejecución de las actividades.

Ahora para acceder a los resultados de Hadoop se debe digitar el siguiente URL en la barra de direcciones: **http://master-node.centos:50070**. El puerto 50070, es el puerto predeterminado para acceder a Hadoop y como ya se ha aclarado anteriormente el nombre que está ubicado antes del puerto es la identificación que se colocó en el archivo ***core-site.xml*** para acceder a los servicios de Hadoop, en otros casos se puede colocar por defecto ***localhost***, todo depende del nombre que se haya colocado en dicho archivo de configuración. De esta manera, se muestran las siguientes imágenes:

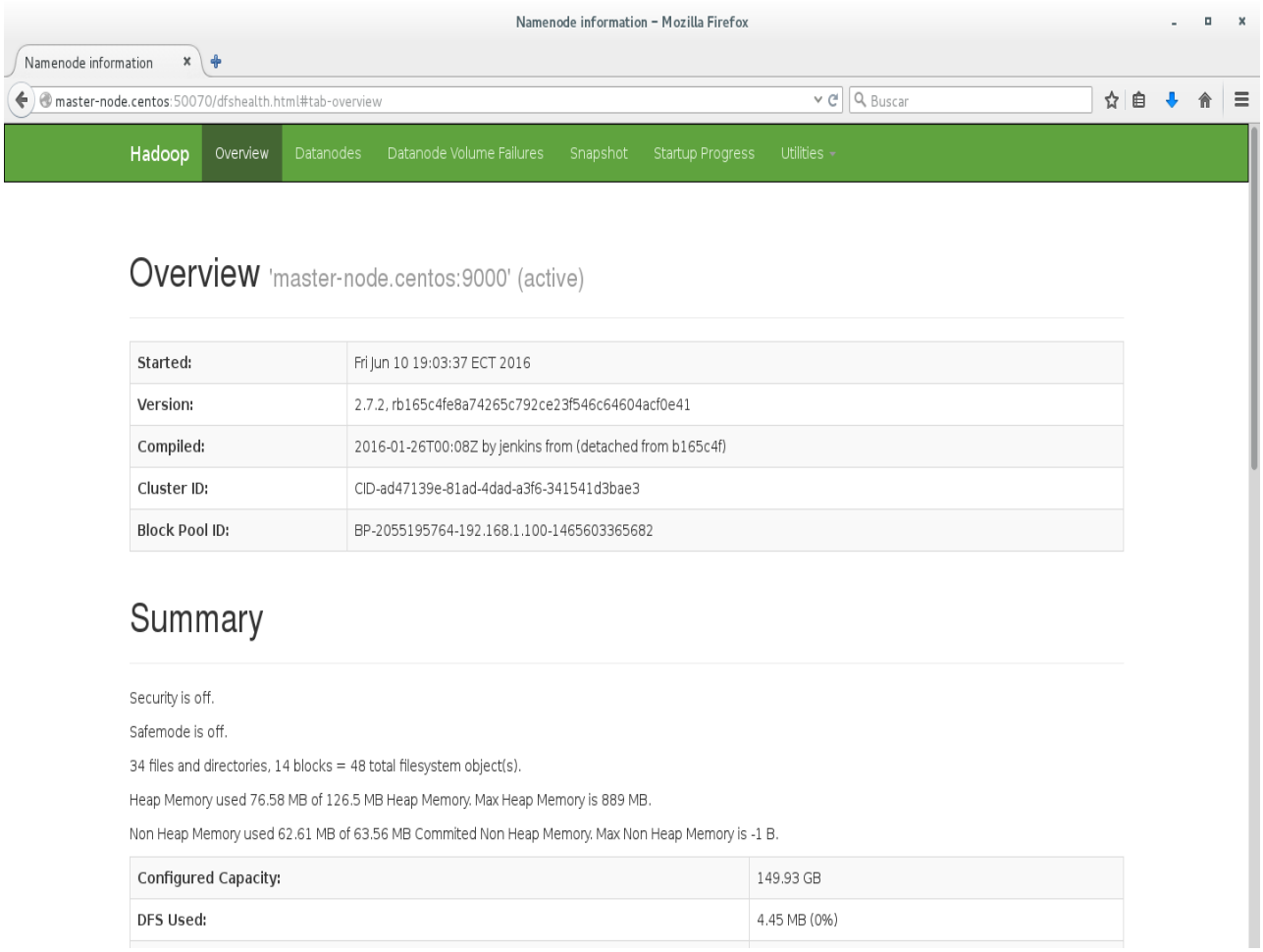


Figura 4.110: Vista en el explorador de la información de Hadoop.

En la imagen anterior, se puede observar la información de Hadoop como su versión, la fecha de inicio de esta aplicación y los recursos que utiliza para su funcionamiento.

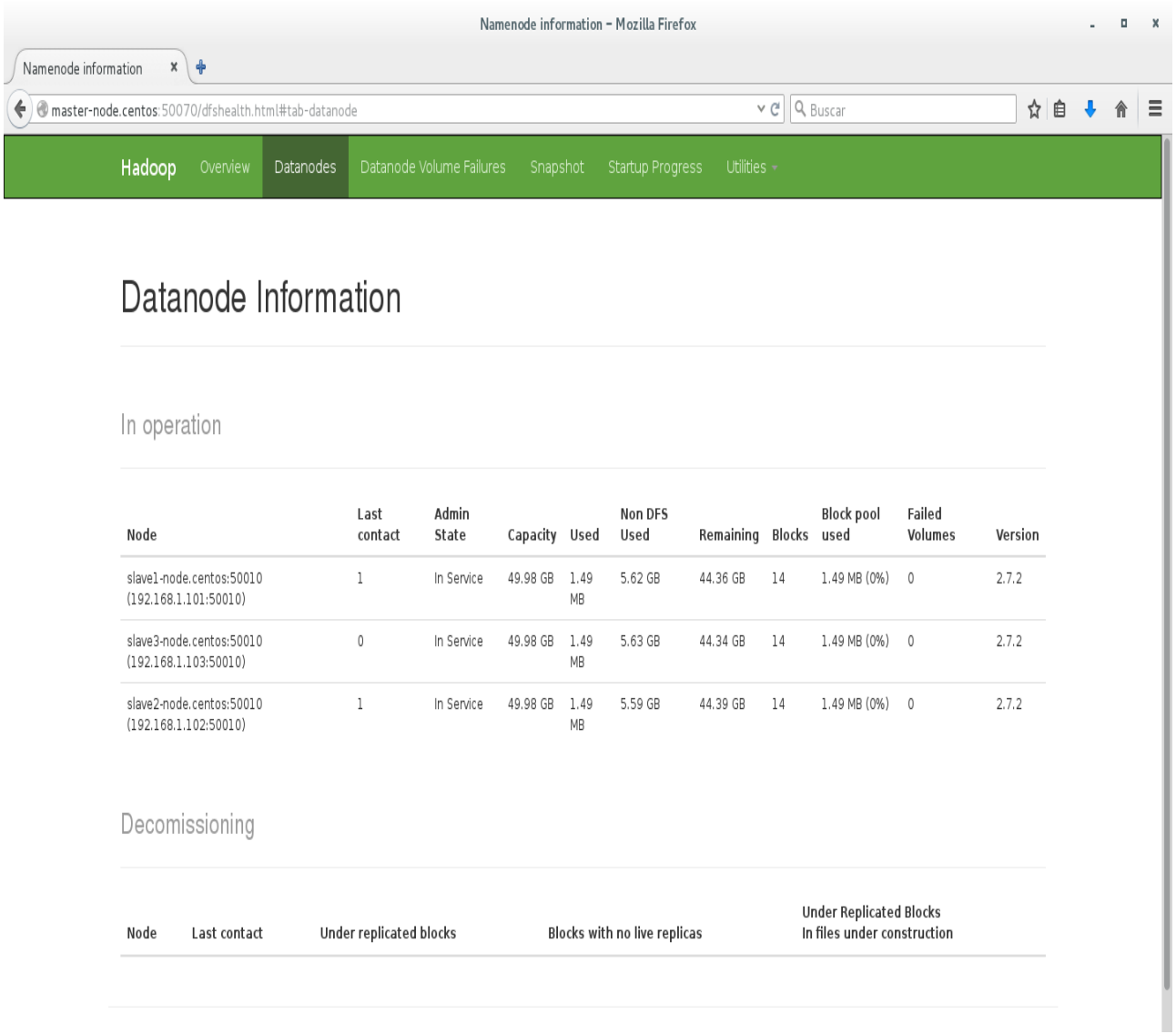


Figura 4.111: Vista en el explorador de la información de Hadoop.

En la imagen anterior, se puede observar la información correspondiente a los Nodos Esclavo como por ejemplo, cuáles están en servicio y los recursos que consume cada uno.

Las siguientes imágenes indican el rendimiento de la máquina durante la ejecución de Hadoop:

Figura 4.112: Rendimiento de la máquina antes de la ejecución de Hadoop.

La figura anterior indica el rendimiento de la máquina antes de la ejecución de Hadoop, se puede observar que los valores se encuentran en un estado normal.

Figura 4.113: Rendimiento de la máquina al inicio de la ejecución de Hadoop.

La imagen anterior muestra que los valores de memoria e intercambio como los valores de monitoreo de la red se colocan en cero al momento de iniciar Hadoop, esto es debido a que Hadoop consume rápidamente los recursos de la máquina.

Figura 4.114: Rendimiento de la máquina durante la ejecución de Hadoop.

La figura anterior muestra que los valores de rendimiento del CPU, Memoria y Red de la máquina cambian considerablemente durante la ejecución de los procesos de Hadoop.

4.2. Casos Prácticos

4.2.1. Ejecución Ejemplo WordCout (Contador de Palabras)

En esta sección se explica el proceso de desarrollo para la ejecución de algoritmos en ambientes de Big Data. Se explica detalladamente los procesos que se debe seguir para ejecución de aplicaciones MapReduce.

Al realizar aplicaciones para Hadoop, se deben desarrollar en el lenguaje de programación Java, en vista que Hadoop fue desarrollado en Java.

El caso práctico escogido consiste en contar palabras de un texto, este ejemplo es una clara representación de MapReduce. Debido a que se realizan tareas "Map", en las cuales se identifican las palabras que contiene el archivo de texto, posteriormente se realizan tareas "Reduce" en las cuales se identifican las palabras que están repetidas y de esta manera se tiene como resultado el número de palabras totales que se encuentran en el archivo de texto analizado.

4.2.1.1. Pasos de Ejecución

PASO 1: Utilizar una plataforma para desarrollar entornos de desarrollo (IDE), la plataforma deber permitir generar archivos .jar.

En caso de no contar con algún IDE de java, se explica a continuación la instalación de Eclipse para sistemas operativos que corresponda a la cadena de Red Hat. Para ello es necesario descargar "Eclipse IDE for Java EE Developers" de la página web de Eclipse (http://www.eclipse.org/downloads/packages/release/Mars/2).

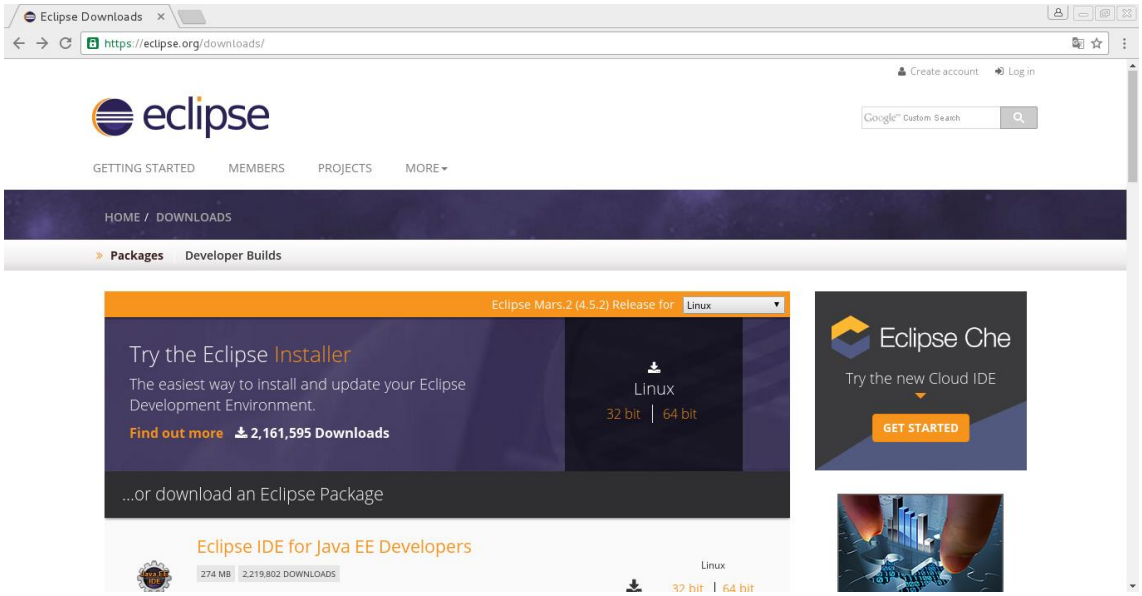


Figura 4.115: Eclipse, página web.

- Iniciar el Terminal, es necesario entrar con privilegios de root.

Digitar los siguientes comandos:

- Para acceder al sistema con privilegios de root:

$su

❖ Una vez descargado Eclipse, es necesario copiar el instalador al directorio */home/principal*, debido a que todas las descargas realizadas se ubican en el directorio *Descargas.*

Digitar los siguientes comandos:

- Para copiar el instalador de Eclipse ubicado en el directorio *Descargas* al directorio */home/principal*:

```
#cp /home/principal/Descargas/eclipse-jee-mars-2-linux-gtk-x86_64.tar.gz
/root/
```

Figura 4.116: Copiar eclipse, a la dirección de usuario.

- Para listar el contenido del directorio:

```
#ls
```

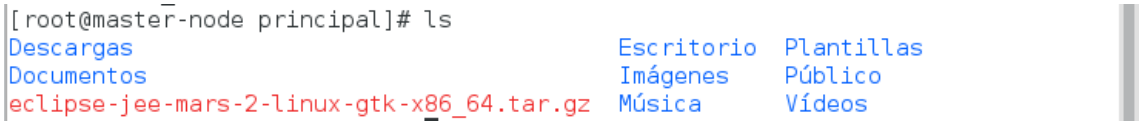


Figura 4.117: Verificar el archivo copiado.

❖ Extraer el instalador en el directorio */opt* :

```
#tar -zxvf eclipse-jee-mars-2-linux-gtk-x86_64.tar.gz -C /opt/
```

Figura 4.118: Descomprimir el archivo.

❖ Realizar un enlace al directorio */bin* :

```
#ln -s /opt/eclipse/eclipse /usr/bin/eclipse
```

Figura 4.119: Enlace a directorio /bin.

- Crear la aplicación o el lanzador Gnome:

```
#ln –s /opt/eclipse/eclipse /usr/bin/eclipse
```

```
[root@master-node principal]# vi /usr/share/applications/eclipse.desktop
```

Figura 4.120: Editar lanzador de Gnome.

- Digitar los siguientes comandos, para crear el lanzador:

```
[Desktop Entry]
Encoding = UTF-8
Name = 4.4.1 Eclipse
Comentario = Eclipse de Luna
Exec = / usr / bin / eclipse
Icono = / opt / eclipse / icon.xpm
Categorías = Aplicación; Desarrollo; Java IDE;
Version = 1,0
Type = Aplicación
Terminal = 0
```

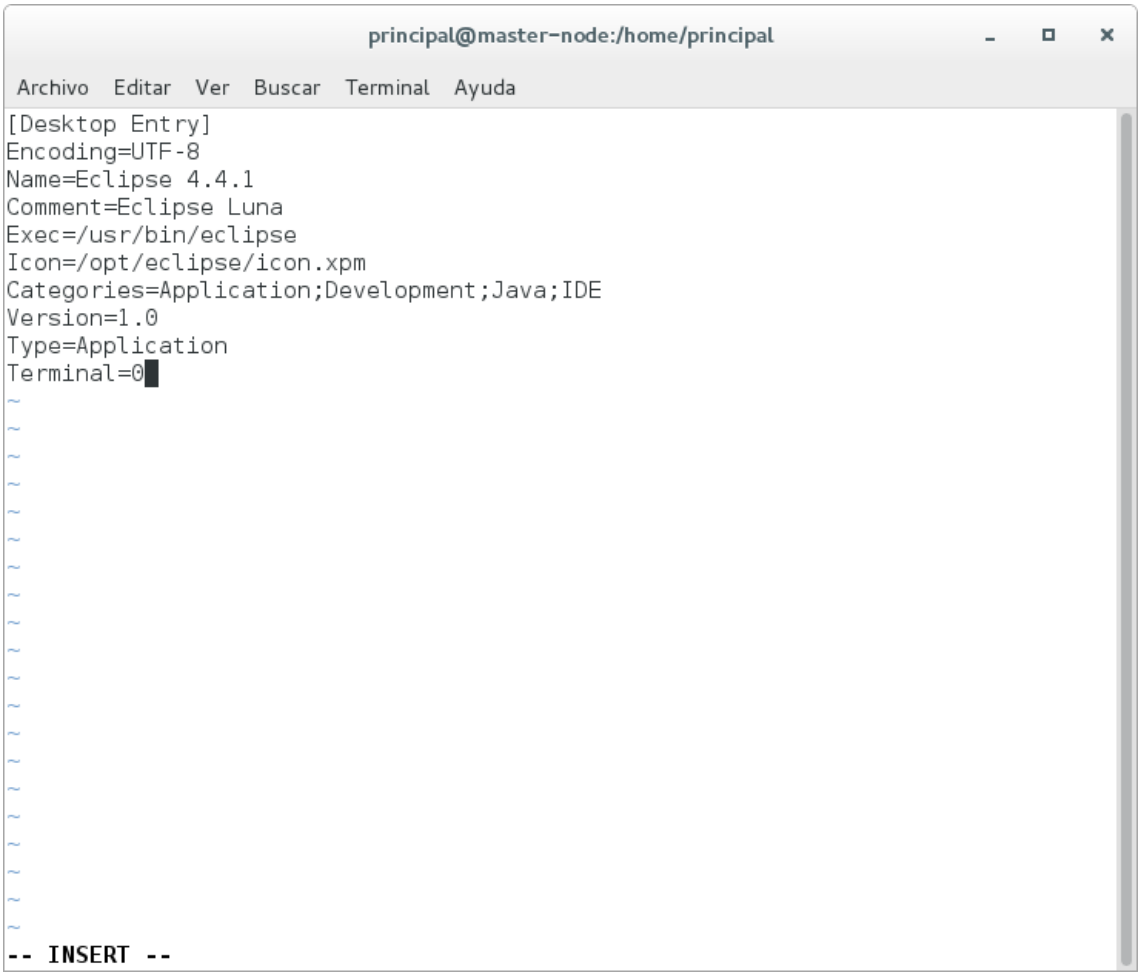


Figura 4.121: Editar lanzador de Gnome con gestor vi.

- Abrir Eclipse:

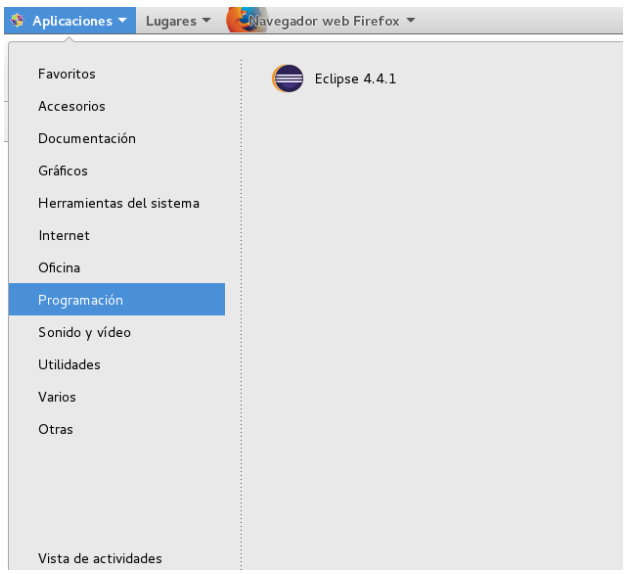


Figura 4.122: Resultado del lanzador.

PASO 2: Creación de una nueva aplicación.

Al desarrollar aplicaciones para Hadoop se debe tener en cuenta ciertos pasos, los mismos que deben ser comprendidos para el desarrollo de cualquier aplicación.

- Iniciar Eclipse, y digitar la ubicación en donde se guardará el proyecto.

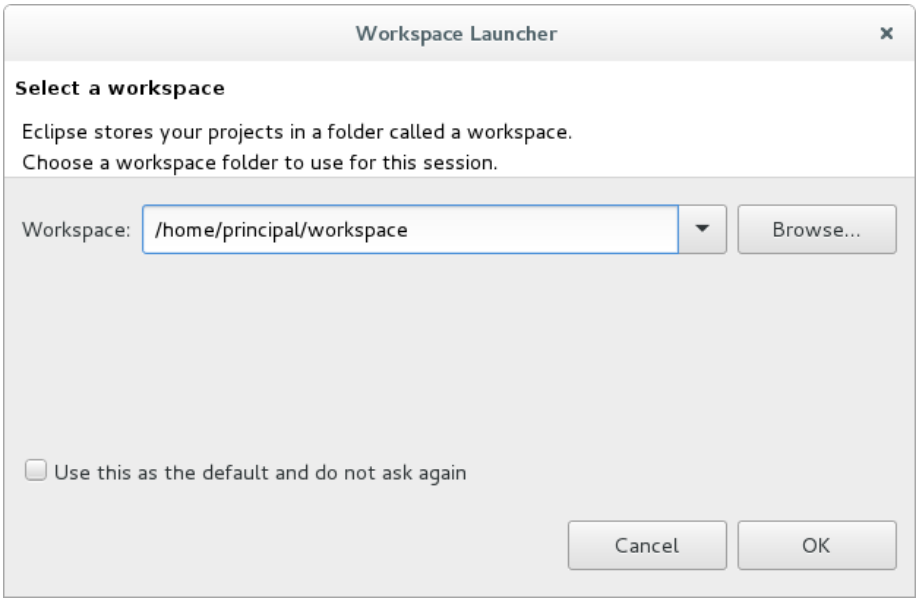


Figura 4.123: Directorio de Eclipse.

- Crear un nuevo proyecto de java. En la siguiente ventana se selecciona el nombre del proyecto y se verifica que el jdk esté por defecto.
- Seleccionar proyecto de java a realizar.

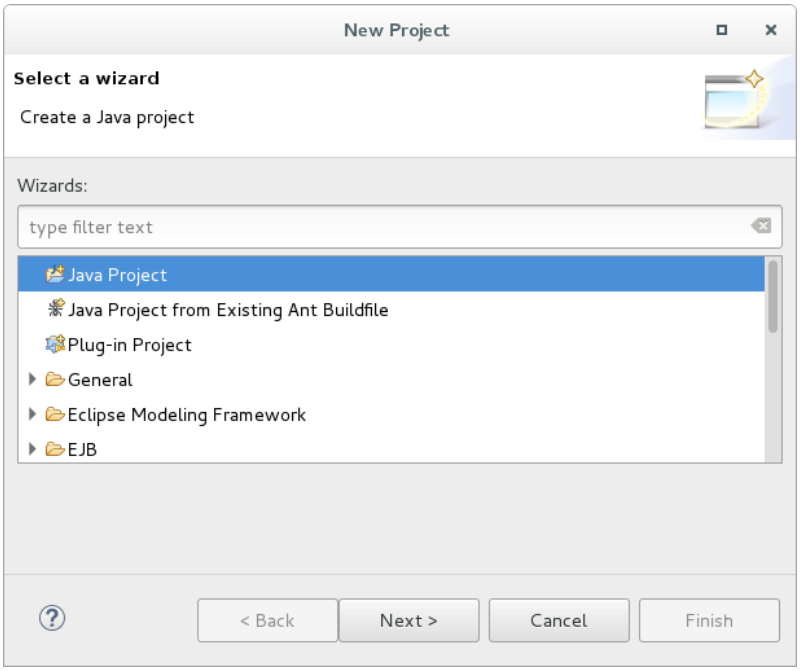


Figura 4.124: Selección de Java Project.

- Escribir el nombre de la aplicación que se desea realizar en este caso, se llamará Ejemplo2WordCount y presionar en el botón finalizar.

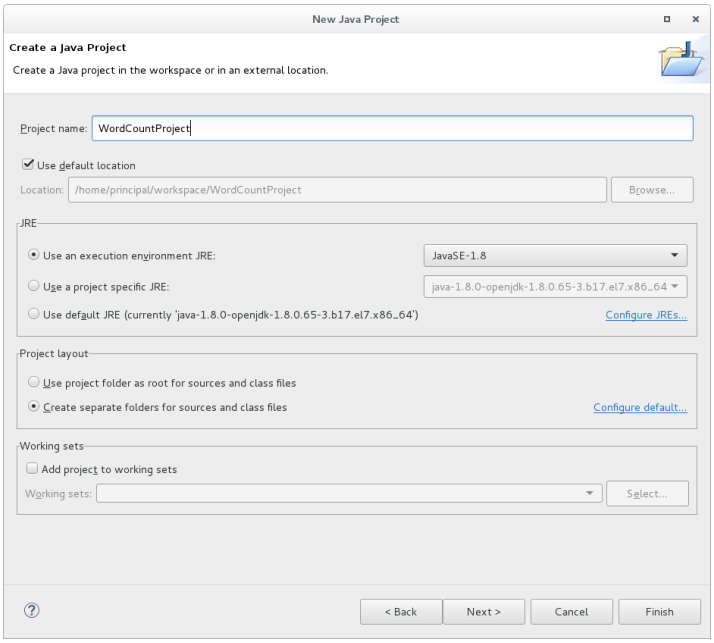


Figura 4.125: Nombre de la aplicación.

PASO 3: Programar la aplicación.

- Para esto es necesario crear una clase, en este caso la clase se denominará WordCount.

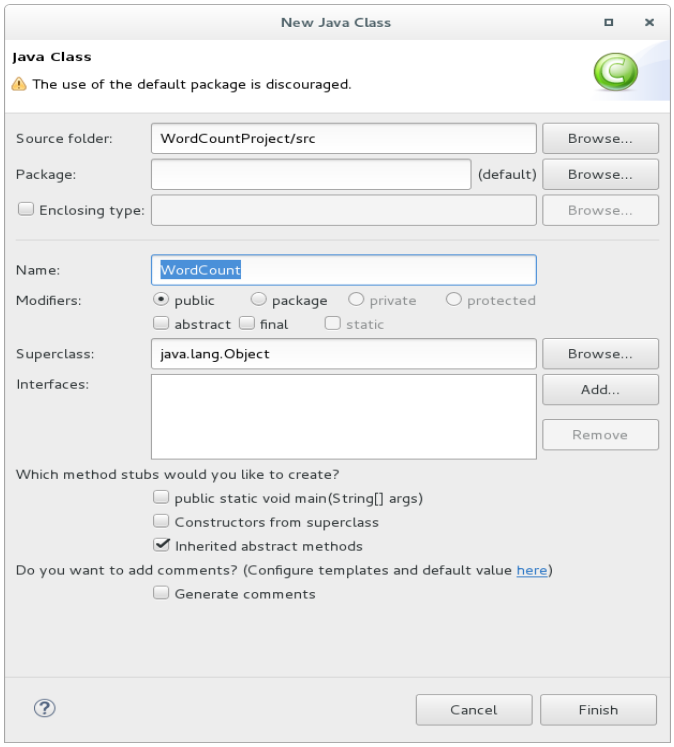


Figura 4.126: Creación de la clase.

Teniendo en cuenta los conceptos mencionados en capítulos anteriores, es necesario desarrollar las tareas “Map”.

```
public static class TokenizerMapper
       extends Mapper<Object, Text, Text, IntWritable>{

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context
                    ) throws IOException, InterruptedException {
      StringTokenizer itr = new StringTokenizer(value.toString());
      while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
      }
    }
  }
```

Teniendo en cuenta los conceptos mencionados en capítulos anteriores, es necesario desarrollar las tareas “Reduce”.

```
  public static class IntSumReducer
       extends Reducer<Text,IntWritable,Text,IntWritable> {
```

```
  private IntWritable result = new IntWritable();
  public void reduce(Text key, Iterable<IntWritable> values,
                     Context context
                     ) throws IOException, InterruptedException {
    int sum = 0;
    for (IntWritable val : values) {
      sum += val.get();
    }
    result.set(sum);
    context.write(key, result);
  }
}
```

Posteriormente se establece el método Main, que contiene todas las llamadas a los Jobs, para que se ejecuten sincronizadamente.

```
public static void main(String[] args) throws Exception {
  Configuration conf = new Configuration();
  Job job = Job.getInstance(conf, "word count");
  job.setJarByClass(WordCount.class);
  job.setMapperClass(TokenizerMapper.class);
  job.setCombinerClass(IntSumReducer.class);
  job.setReducerClass(IntSumReducer.class);
  job.setOutputKeyClass(Text.class);
  job.setOutputValueClass(IntWritable.class);
  FileInputFormat.addInputPath(job, new Path(args[0]));
  FileOutputFormat.setOutputPath(job, new Path(args[1]));
  System.exit(job.waitForCompletion(true) ? 0 : 1);
}
```

Finalmente, se debe importar las librerías de Hadoop, existen varias librerías con diferentes funciones, para este caso particular se deben agregar las librerías de la carpeta common y mapreduce.

- Click derecho en el proyecto, posteriormente se escoge la opción *Build Path y* después Click en *Configure Build Path…*

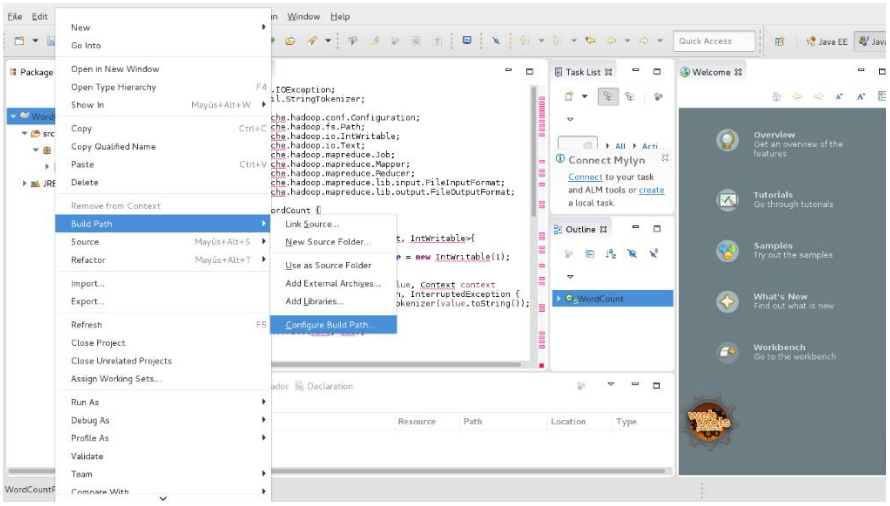


Figura 4.127: Importar librerías.

❖ Seleccionar *Add External JARs*, y agregar las librerías siguientes.

- hadoop-common-2.7.2-tests.jar
- hadoop-common-2.7.2.jar
- hadoop-mapreduce-client-app-2.7.2.jar
- hadoop-mapreduce-client-common-2.7.2.jar
- hadoop-mapreduce-client-core-2.7.2.jar
- hadoop-mapreduce-client-hs-2.7.2.jar
- hadoop-mapreduce-client-hs-plugins-2.7.2.jar
- hadoop-mapreduce-client-jobclient-2.7.2-tests.jar
- hadoop-mapreduce-client-jobclient-2.7.2.jar
- hadoop-mapreduce-client-suffle-2.7.2.jar
- hadoop-mapreduce-examples-2.7.2.jar
- hadoop-nfs-2.7.2.jar

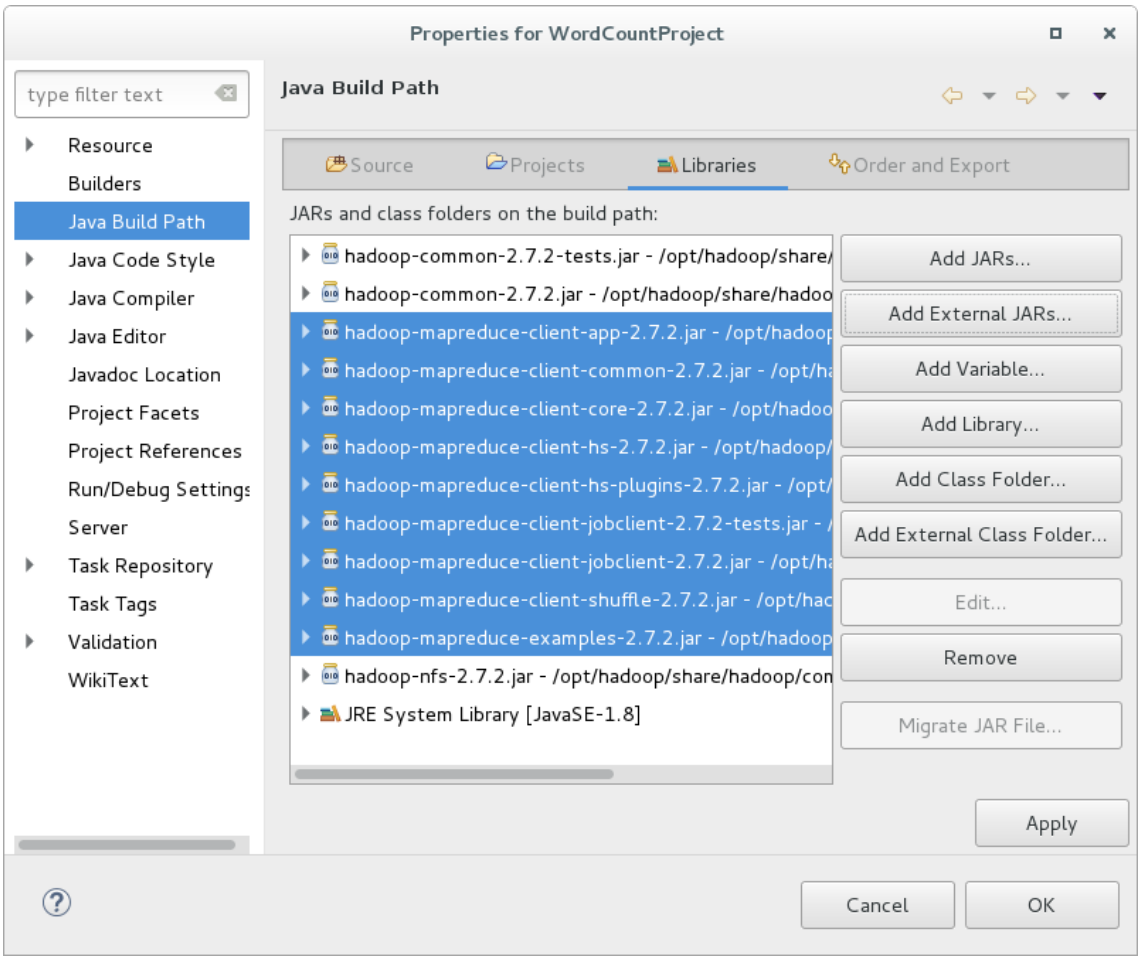


Figura 4.128: Importar librerías jar externas.

Al agregar las librerías automáticamente se eliminan los errores, pero al ejecutar el programa se muestran ciertas alertas, estas alertas muestran información superficial que no indican de manera clara si el programa se realizó con éxito, para comparar que la aplicación ha sido desarrollada con éxito es necesario correr el archivo .jar en Hadoop.

Como resultado final se obtiene lo siguiente:

```
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {

  public static class TokenizerMapper
       extends Mapper<Object, Text, Text, IntWritable>{

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context
                    ) throws IOException, InterruptedException {
      StringTokenizer itr = new StringTokenizer(value.toString());
      while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
      }
    }
  }
  public static class IntSumReducer
       extends Reducer<Text,IntWritable,Text,IntWritable> {
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<IntWritable> values,
                       Context context
                       ) throws IOException, InterruptedException {
      int sum = 0;
      for (IntWritable val : values) {
        sum += val.get();
      }
      result.set(sum);
      context.write(key, result);
    }
  }
  public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "word count");
    job.setJarByClass(WordCount.class);
```

```
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

PASO 4: Generar archivo .jar

Para finalizar el desarrollo de la aplicación, es necesario crear un archivo .jar, este archivo contiene toda las líneas de código programadas. Para esto se debe seguir los siguientes pasos.

- Click derecho en el proyecto, buscar la opción Export.
- Seleccionar tipo de exportación, escoger *Jar file.*

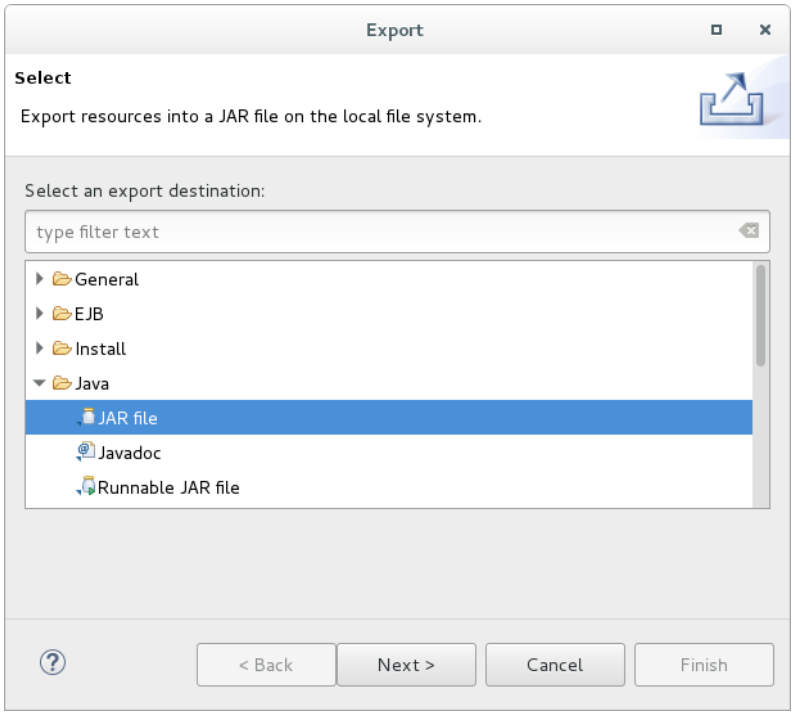


Figura 4.129: Destino de Exportación JAR file.

- Seleccionar la dirección en donde se desea guardar el archivo .jar.

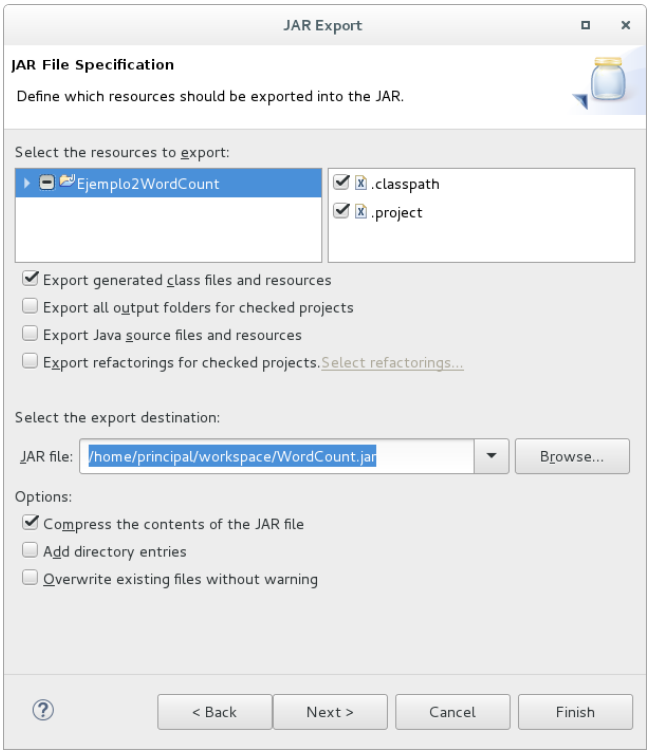


Figura 4.130: Guardar .jar en la dirección establecida.

PASO 5: Ejecución de la aplicación

Para la ejecución de este caso práctico, se requiere un archivo de texto plano, el contenido puede ser generado en cualquier editor de texto, para este ejemplo se utilizó Gedit y debe ser guardaba en la misma dirección en la que se guarda el archivo **.jar**.

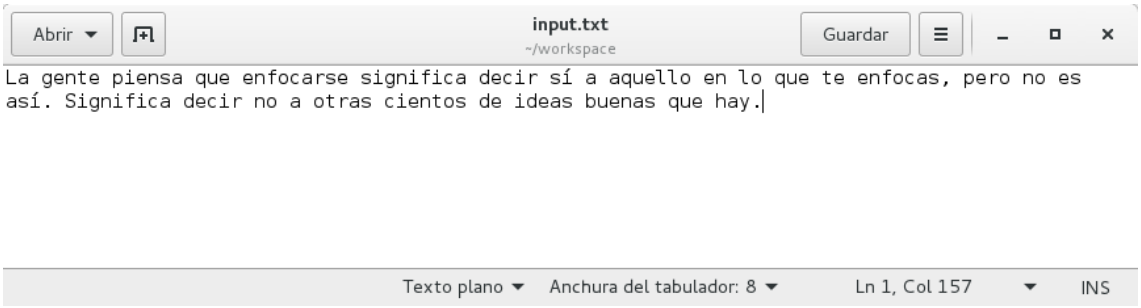


Figura 4.131: Archivo de texto.

- Verificar el Almacenamiento de los archivos.

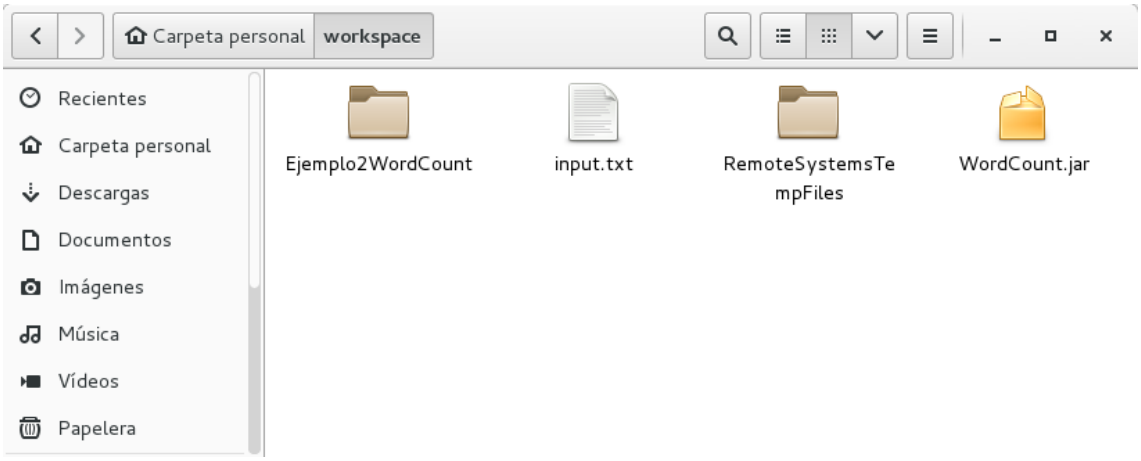


Figura 4.132: Verificar el almacenamiento de los archivos.

- Abrir la terminal y acceder con privilegios de root.

- Entrar a la carpeta donde se encuentra los archivos requeridos.

```
# cd /home/principal/workspace
```

```
[root@master-node principal]# cd /home/principal/workspace/
```

Figura 4.133: Acceder a la carpeta que contiene archivos

- Copiar el archivo input.txt a la carpeta del sistema *hdfs.*

```
# hdfs dfs –moveFromLocal input.txt /inputWordCount
```

```
[root@master-node workspace]# hdfs dfs -moveFromLocal input.txt /inputWordCount
16/07/03 20:11:46 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
```

Figura 4.134: Copiar archivo a sistema hdfs.

Antes de continuar con los pasos subsiguientes es necesario explicar qué es y cómo funciona la instrucción *HDFS.*

HDFS (Hadoop Distribuid File System) es un sistema de ficheros distribuido, portátil y escalable. Una de las características principales es el tamaño de un bloque muy superior a los 64 MB, esto resulta muy útil al momento de realizar accesos de lectura ya que no se pierde tiempo en acceder.

Por lo general, los ficheros que van a ser almacenados y ubicados en este tipo de ficheros de Hadoop, siguen el mismo patrón “Write once read many” (escribe una vez y lee muchas veces). Pero estos permisos pueden variar de acuerdo a la necesidad del usuario, es así que en algunos casos se utiliza el comando *chmod.*

Finalmente, los ficheros serán divididos en bloques del mismo tamaño y distribuidos entre los nodos que conforman el clúster.

Como se había mencionado en capítulos anteriores existen varios tipos de demonios con los que cuenta Hadoop, HDFS trabaja exclusivamente con dos demonios los cuales son: Namenode y Datanode.

- Se debe comprobar que el archivo de texto forma parte del fichero hdfs de Hadoop, para ello se ejecuta el siguiente comando:

```
# hdfs dfs –ls /inputWordCount
```

```
[root@master-node workspace]# hdfs dfs -ls /inputWordCount
16/07/03 20:12:18 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
-rw-r--r--   1 root supergroup        159 2016-07-03 20:11 /inputWordCount
```

Figura 4.135: Consulta del fichero con permisos -rw-r--r--.

- Realizada la verificación de archivo de texto, se procede a realizar la ejecución, para eso se utiliza la siguiente estructura.

```
# hadoop jar WordCount.jar WordCount /inputWordCount /outputWC
```

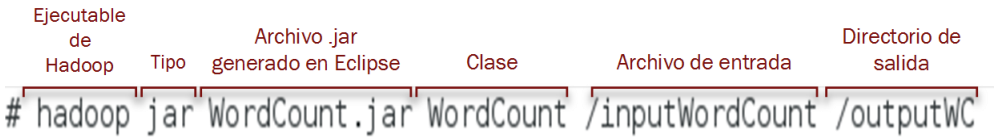


Figura 4.136: Ejecución en Hadoop con archivos de texto.

- Al ejecutar la secuencia se realiza las tareas map y reduce.

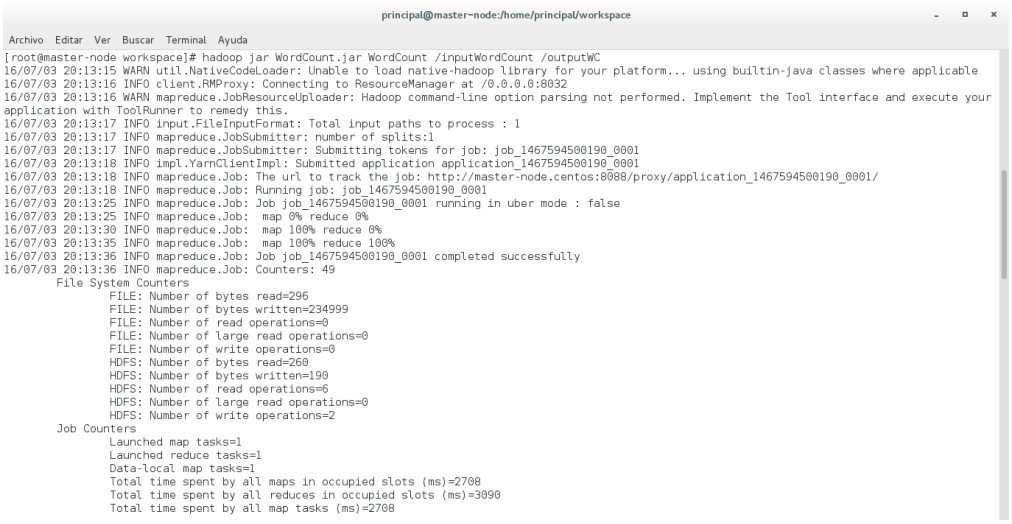


Figura 4.137: Resultado de operación.

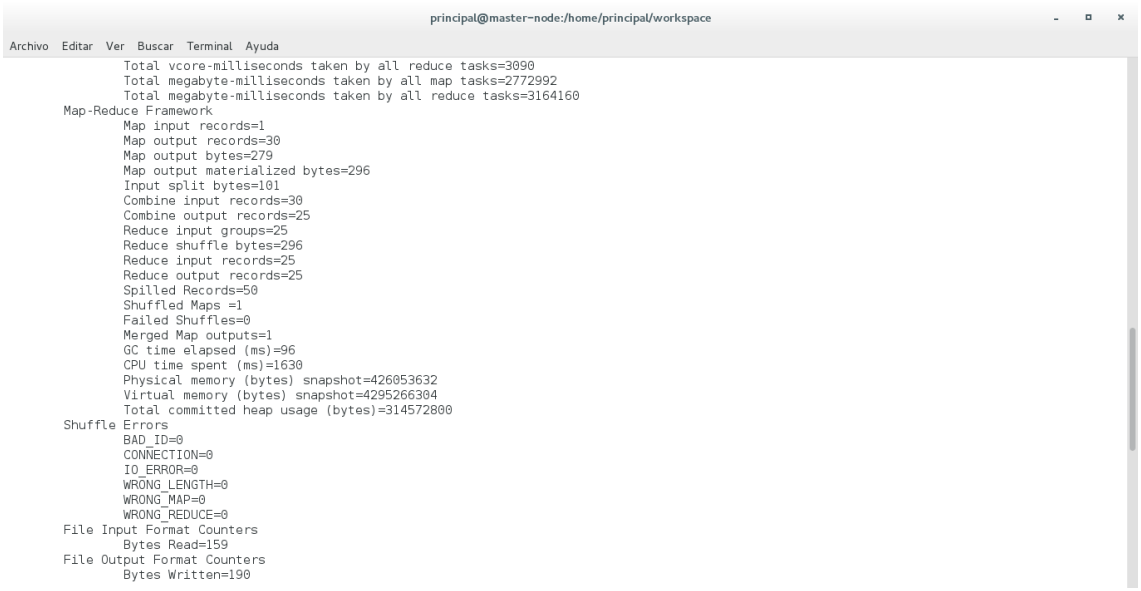


Figura 4.138: Resultado de operación 2.

- Finalmente, Hadoop muestra que proceso fue desarrollado con éxito. Pero para ver el resultado y número de palabras contadas es necesario el siguiente comando.

```
# hdfs dfs -cat /outputWC/*
```

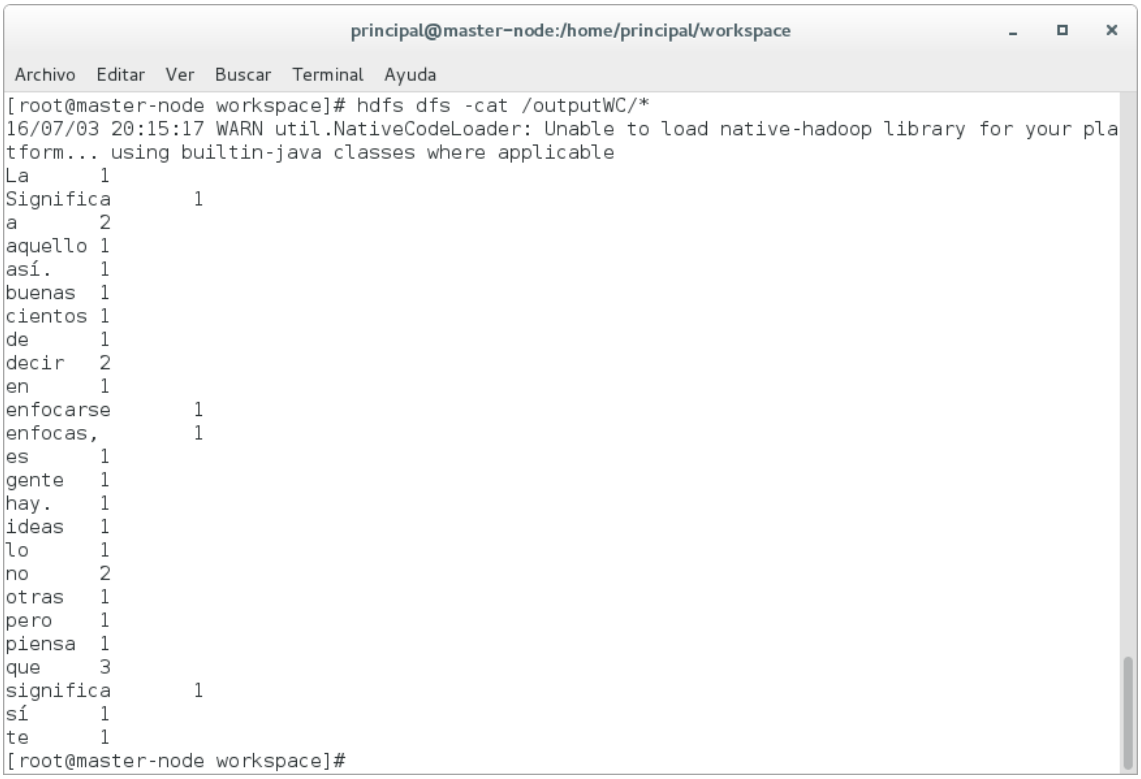


Figura 4.139: Resultado final Word Count.

4.2.2. Recolección de Tweets en una Base de Datos NoSQL.

Actualmente existen varias tipos de redes sociales como Twitter, Facebook, Instagram, Google+, etc. En los últimos años las redes sociales se han convertido en los sitios web más visitados en internet, donde las personas opinan, comparten ideas y se conectan por diversión y también con fines educativos. Las redes sociales diariamente manejan grandes volúmenes de datos e información. Para este caso práctico se utilizará Twitter y se recolectará Tweets de una determinada zona geográfica y se almacenará en una base de datos NoSQL.

Antes de realizar esta práctica es importante mencionar algunos conceptos, que serán de gran ayuda al momento de realizar este caso práctico.

“Twitter es una red de información conformada por mensajes de 140 caracteres llamados Tweets. Es una forma fácil de descubrir las últimas novedades relacionadas con los temas que te interesan.” (Twitter, 2016)

Por otro lado, los Tweets están conformados por texto, hashtags o etiquetas para categorizar, @nombre_de_usuario y en algunos casos pueden adjuntar una dirección URL, tal como se ilustra en la siguiente figura.

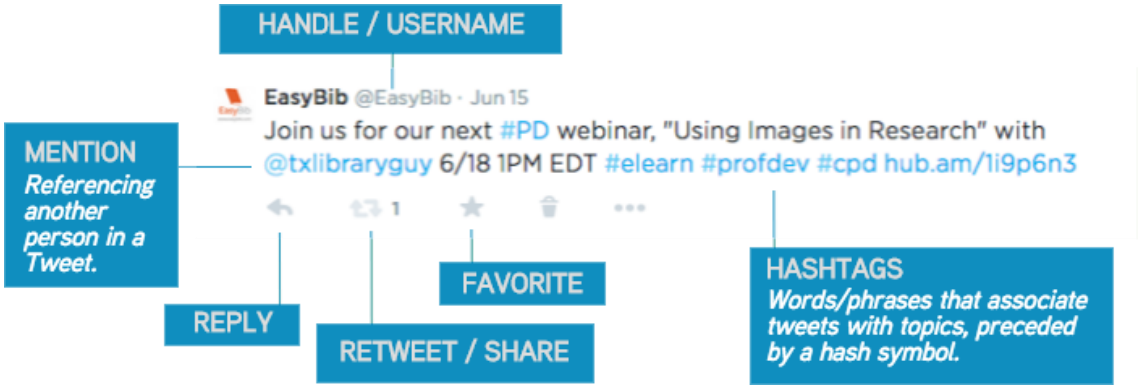


Figura 4.140: Estructura de Tweets.

La estructura es la siguiente:

- Texto o mensaje que se quiere compartir con esta red de información, puede estar compuesto por 140 caracteres y su publicación es inmediata.
- Hashtag o etiquetas para categorizar, es un símbolo # seguido de un texto que permite aumentar audiencia de un Tweet, son utilizados para mencionar ciertas acciones, eventos, sentimientos, etc. Estas etiquetas pueden ser utilizadas en el análisis de datos.
- **Dirección URL abreviada**, estos enlaces son opcionales y son utilizados para montar cierto contenido externo.
- @ **Nombre de usuario.** Es una identificación obligatoria usada para reconocer a los usuarios.

Diariamente se generan gran cantidad de Tweets que son compartidos por los usuarios y contienen datos que si son analizados podrían generar información, es así que Twitter ofrece APIs, que permiten a los desarrolladores adaptarse a diferentes necesidades. Por ejemplo, existe el Streaming API mismo que permite el acceso en tiempo real a los Tweets que han sido publicados, es decir, se crea una conexión permanente a través del usuario con los servidores de Twitter recibiéndose un flujo constante de Tweets en formato **Json**[12]; el Rest

[12] **JSON** (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - Diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl,

API permite que los desarrolladores puedan acceder al núcleo central donde se encuentran los datos de Twitter y, por último, se encuentra el Search API el cual ofrece una información mucho más limitada de los Tweets, permitiendo solo el acceso a los datos del autor como el id, el nombre del usuario con el que aparece en Twitter, tanto el Streaming API como el REST API permiten acceder al perfil completo del autor.

Para poder utilizar Twitter en un ambiente de desarrollo es necesario tener una cuenta de Twitter previamente creada.

En este caso en particular se realizará la práctica de recolección de Tweets como una demostración de lo que es Big Data, es decir, la recolección de grandes de volúmenes de datos.

Para llevar a cabo esta práctica se utilizarán las siguientes herramientas:

- Lenguaje de programación Python.
- Base de Datos NoSQL como CouchDB.
- Sistema operativo compatible con CouchDB, como Ubuntu 16.04 LTS.

A continuación se muestran los pasos que se deben seguir para llevar a cabo esta práctica:

Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.**Fuente especificada no válida.**

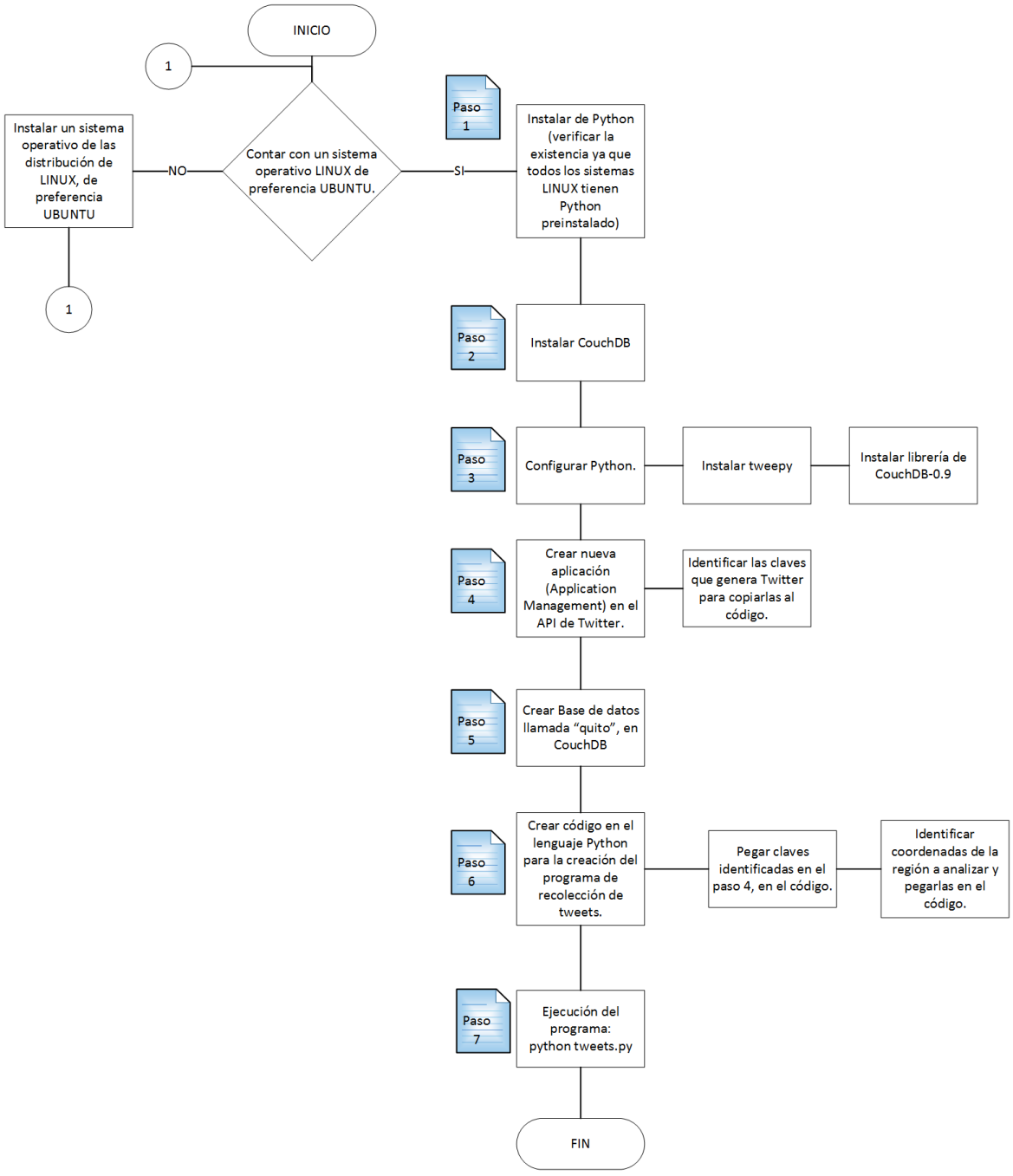


Figura 4.141: Diagrama de flujo del proceso de recolección de Tweets.

PASO 1: Instalación de Python

Los sistemas operativos que corresponden a la cadena de distribución GNU/Linux tienen ya instalado Python por defecto, por lo cual no es necesario realizar algún tipo de instalación para este caso.

PASO 2: Instalación de CouchDB.

"CouchDB conocida oficialmente como Apache CouchDB, es una base de datos orientada a documentos, la cual pertenece a las bases de datos NoSQL. Puede ser consultada e indexada usando JavaScript como función MapReduce. CouchDB ofrece una API (Application Programming Interface) RESTful (Representational State Transfer) en JSON que puede ser accedida vía peticiones HTTP. Existen muchas librerías para casi cualquier lenguaje de programación que facilitan el acceso. CouchDB está escrito en Erlang, un lenguaje de programación funcional robusto ideal para construir sistemas distribuidos simultáneos, lo que permite un diseño flexible y fácilmente escalable y extensible." (Garcia, 2009)

Actualmente, en la versión de 16.04 de Ubuntu la instalación de ciertos programas es mucho más sencilla puesto que no se necesita de la instalación de ciertos paquetes del sistema, hecho que no sucede en versiones más antiguas de Ubuntu en donde sí es necesaria la instalación de ciertos paquetes para que los programas funcionen correctamente.

- Se requiere instalar la versión más reciente de CouchDB, para esto se utilizan los repositorios **ppa** que son los que permiten adquirir la versión actualizada de cualquier programa, se digita entonces el siguiente comando:

```
$sudo add-apt-repository ppa:couchdb/stable -y
```

```
bigdata@bigdata:~$ sudo add-apt-repository ppa:couchdb/stable -y
gpg: anillo «/tmp/tmpdrz8yk5g/secring.gpg» creado
gpg: anillo «/tmp/tmpdrz8yk5g/pubring.gpg» creado
gpg: solicitando clave C17EAB57 de hkp servidor keyserver.ubuntu.com
gpg: /tmp/tmpdrz8yk5g/trustdb.gpg: se ha creado base de datos de confianza
gpg: clave C17EAB57: clave pública "Launchpad PPA for Apache CouchDB" importada
gpg: Cantidad total procesada: 1
gpg:               importadas: 1  (RSA: 1)
OK
```

Figura 4.142: Instalación repositorio actual de CouchDB.

- Realizar una actualización de la lista en caché de los paquetes.

```
$sudo apt-get update
```

```
bigdata@bigdata:~$ sudo apt-get update
[sudo] password for bigdata:
Obj:1 http://security.ubuntu.com/ubuntu xenial-security InRelease
Obj:2 http://ec.archive.ubuntu.com/ubuntu xenial InRelease
Obj:3 http://ec.archive.ubuntu.com/ubuntu xenial-updates InRelease
Obj:4 http://ec.archive.ubuntu.com/ubuntu xenial-backports InRelease
AppStream cache update completed, but some metadata was ignored due to errors.
Leyendo lista de paquetes... Hecho
```

Figura 4.143: Actualización de los paquetes.

❖ Eliminar cualquier posible existencia de los binarios de CouchDB.

```
$sudo apt-get remove couchdb couchdb-bin couchdb-common -yf
```

```
bigdata@bigdata:~$ sudo apt-get remove couchdb couchdb-bin couchdb-common -yf
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
El paquete «couchdb» no está instalado, no se eliminará
El paquete «couchdb-bin» no está instalado, no se eliminará
El paquete «couchdb-common» no está instalado, no se eliminará
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 260 no actualizados.
```

Figura 4.144: Eliminar binarios de CouchDB.

❖ Instalar CouchDB.

```
$sudo apt-get install –V couchdb
```

```
bigdata@bigdata:~$ sudo apt-get install -V couchdb
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
   couchdb-bin (1.6.1-0ubuntu6ppa2~xenial1)
   couchdb-common (1.6.1-0ubuntu6ppa2~xenial1)
   erlang-asn1 (1:18.3-dfsg-1ubuntu3)
   erlang-base-hipe (1:18.3-dfsg-1ubuntu3)
   erlang-crypto (1:18.3-dfsg-1ubuntu3)
   erlang-eunit (1:18.3-dfsg-1ubuntu3)
   erlang-inets (1:18.3-dfsg-1ubuntu3)
   erlang-mnesia (1:18.3-dfsg-1ubuntu3)
   erlang-os-mon (1:18.3-dfsg-1ubuntu3)
   erlang-public-key (1:18.3-dfsg-1ubuntu3)
   erlang-runtime-tools (1:18.3-dfsg-1ubuntu3)
   erlang-snmp (1:18.3-dfsg-1ubuntu3)
   erlang-ssl (1:18.3-dfsg-1ubuntu3)
   erlang-syntax-tools (1:18.3-dfsg-1ubuntu3)
   erlang-tools (1:18.3-dfsg-1ubuntu3)
   erlang-webtool (1:18.3-dfsg-1ubuntu3)
   erlang-xmerl (1:18.3-dfsg-1ubuntu3)
   libmozjs185-1.0 (1.8.5-1.0.0+dfsg-4.5)
   libsctp1 (1.0.16+dfsg-3)
Paquetes sugeridos:
   erlang (1:18.3-dfsg-1ubuntu3)
   erlang-manpages (1:18.3-dfsg-1ubuntu3)
   erlang-doc (1:18.3-dfsg-1ubuntu3)
```

Figura 4.145: Instalar CouchDB

❖ Detener el servicio de CouchDB, a la vez encenderlo y verificar su estado.

- Para detener el servicio de CouchDB digitar lo siguiente:

```
$sudo systemctl stop couchdb
```

```
bigdata@bigdata:~$ sudo systemctl stop couchdb
```

Figura 4.146: Para CouchDB

- Para encender el servicio de CouchDB digitar lo siguiente:

```
$sudo systemctl start couchdb
```

```
bigdata@bigdata:~$ sudo systemctl start couchdb
```

Figura 4.147: Arrancar servicio de CouchDB.

- Para visualizar el estado de CouchDB digitar lo siguiente:

```
$sudo systemctl status couchdb
```

```
bigdata@bigdata: ~
bigdata@bigdata:~$ sudo systemctl status couchdb
● couchdb.service - Apache CouchDB
   Loaded: loaded (/lib/systemd/system/couchdb.service; enabled; vendor preset:
   Active: active (running) since mié 2016-07-06 15:10:59 ECT; 33s ago
 Main PID: 3651 (beam)
    Tasks: 6 (limit: 512)
   CGroup: /system.slice/couchdb.service
           ├─3651 /usr/lib/erlang/erts-7.3/bin/beam -Bd -K true -A 4 -- -root /u
           └─3666 sh -s disksup

jul 06 15:10:59 bigdata systemd[1]: Started Apache CouchDB.
jul 06 15:10:59 bigdata couchdb[3651]: Apache CouchDB 1.6.1 (LogLevel=info) is s
jul 06 15:10:59 bigdata couchdb[3651]: Apache CouchDB has started. Time to relax
jul 06 15:10:59 bigdata couchdb[3651]: [info] [<0.33.0>] Apache CouchDB has star
jul 06 15:11:13 bigdata systemd[1]: Started Apache CouchDB.
lines 1-14/14 (END)...skipping...
● couchdb.service - Apache CouchDB
   Loaded: loaded (/lib/systemd/system/couchdb.service; enabled; vendor preset:
   Active: active (running) since mié 2016-07-06 15:10:59 ECT; 33s ago
 Main PID: 3651 (beam)
    Tasks: 6 (limit: 512)
   CGroup: /system.slice/couchdb.service
           ├─3651 /usr/lib/erlang/erts-7.3/bin/beam -Bd -K true -A 4 -- -root /u
           └─3666 sh -s disksup

jul 06 15:10:59 bigdata systemd[1]: Started Apache CouchDB.
jul 06 15:10:59 bigdata couchdb[3651]: Apache CouchDB 1.6.1 (LogLevel=info) is s
jul 06 15:10:59 bigdata couchdb[3651]: Apache CouchDB has started. Time to relax
jul 06 15:10:59 bigdata couchdb[3651]: [info] [<0.33.0>] Apache CouchDB has star
jul 06 15:11:13 bigdata systemd[1]: Started Apache CouchDB.
~
~
~
~
~
~
~
~
~
~
lines 1-14/14 (END)...skipping...
● couchdb.service - Apache CouchDB
```

Figura 4.148: Verificar si se encuentra encendido CouchDB.

❖ Finalmente ingresar al explorador y digitar la siguiente dirección:

http://127.0.0.1:5984/_utils/

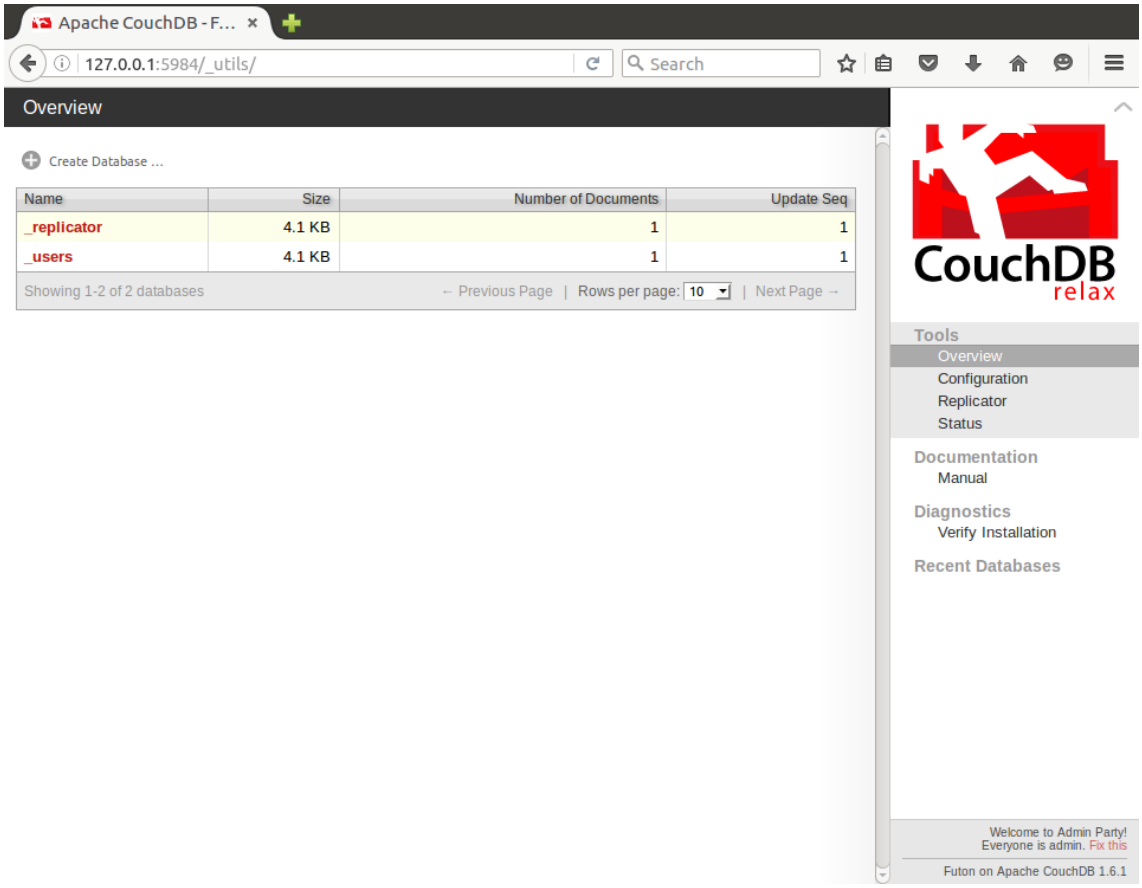


Figura 4.149: Acceso a CouchDB y a sus funciones.

Al instalar CouchDB el bind_andress por defecto es 127.0.0.1, si se desea cambiar esta dirección se puede colocar 0.0.0.0 para poder acceder por medio de localhost. Además, el puerto que inicia CouchDB es por defecto el 5984.

PASO 3: Configuración de Python

Python cuenta con una variedad de librerías con diferentes funciones que son útiles al momento de realizar programas con diferentes plataformas. Es así, que esta práctica debe utilizar el paquete de Python con conexión a CouchDB denominado *CouchDB-0.9*. Por otro lado Python es un lenguaje de programación muy activo en donde existen comunidades de desarrolladores que crean librerías para varios servicios, actualmente existe una librería alojada en GitHub que permite comunicarse con la plataforma de Twitter denominada tweepy. Este repositorio será instalado para este caso práctico.

- Instalar tweepy, el cual permitirá acceder a las llaves generadas por Twitter y así se podrán recolectar los Tweets, hecho que será explicado en pasos posteriores. Para esto es necesario acceder con privilegios de root.

```
#pip install tweepy
```

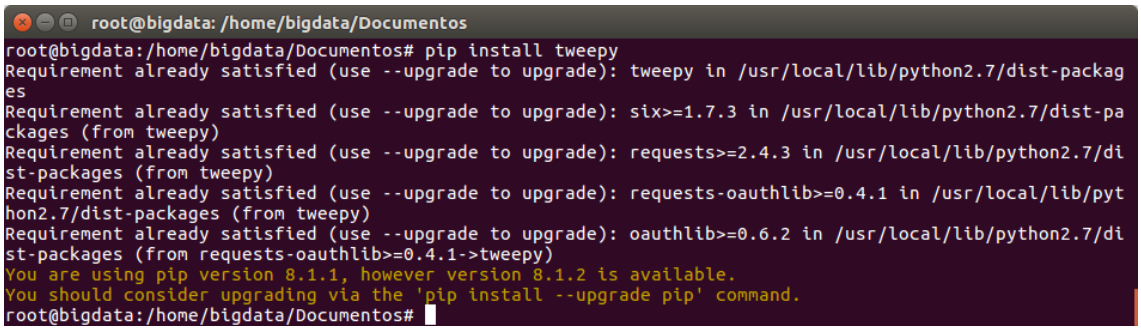


Figura 4.150: Instalar tweepy.

- Descargar y descomprimir la librería CouchDB-09, esta permitirá acceder a CouchDB, desde el código desarrollado en Phyton. Para esto no es necesario estar con privilegios de root.

```
$wget https://pypi.python.org/packages/source/C/CouchDB/CouchDB-0.9.tar.gz
```

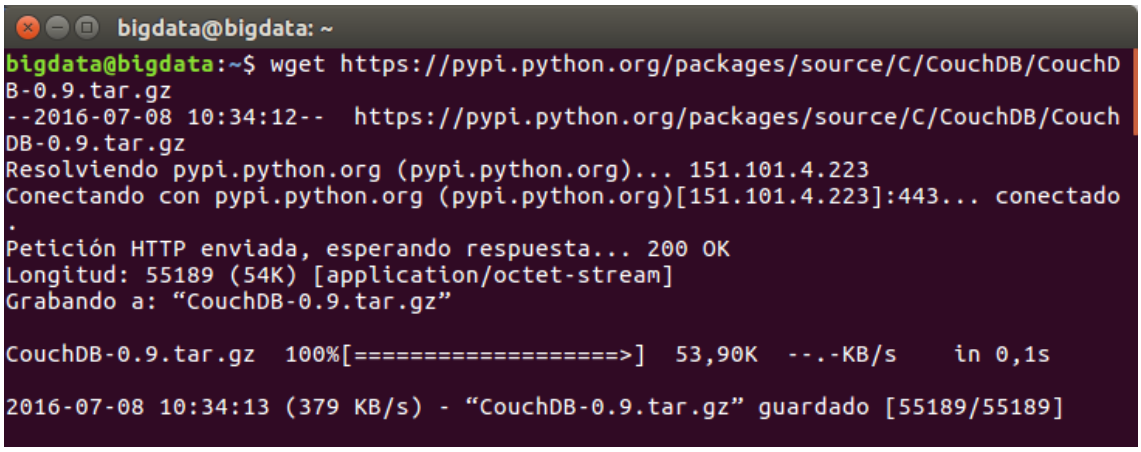


Figura 4.151: Descargar librería CouchDB-09.

- Descomprimir la librería con el siguiente comando.

```
$tar zxvf CouchDB-0.9.tar.gz
```

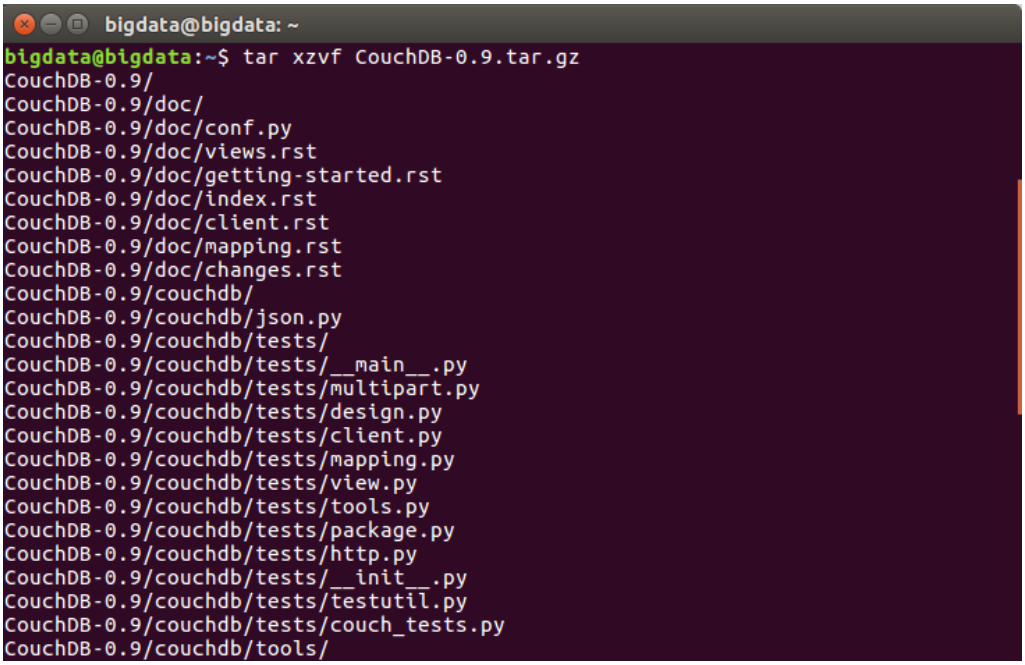


Figura 4.152: Descomprimir librería.

- Acceder al directorio en donde se descomprimió el archivo, como se ilustra a continuación.

```
$cd /home/bigdata/CouchDB-0.9/
```

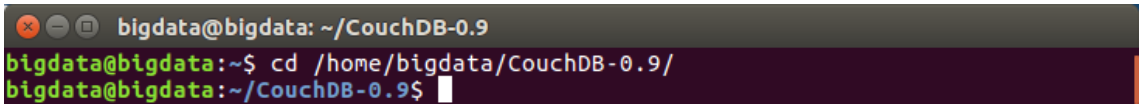


Figura 4. 153: Acceso a carpeta CouchDB-09.

- En el mismo directorio accedido, es necesario instalar archivos que correspondan a la librería descargada, pero deben ser ejecutados con Python para que puedan ser reconocidos por este lenguaje de programación.

```
$python setup.py install
```

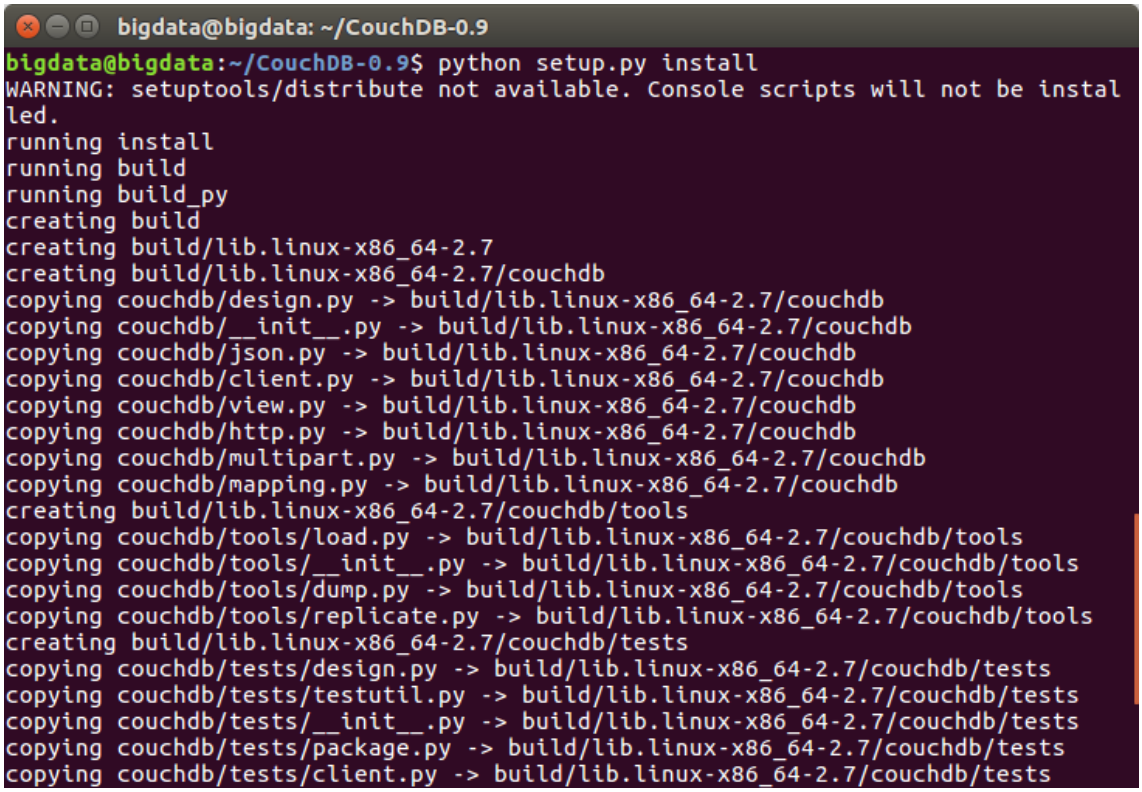


Figura 4.154: Instalación de librerías para que puedan ser reconocidas por python.

- Se debe acceder a Python e importar CouchDB, para esto se necesita digitar el siguiente comando.

- Acceder a Python e importar CouchDB, como se muestra en la figura siguiente:

```
Python

   >>> import couchdb
   >>>exit()
```

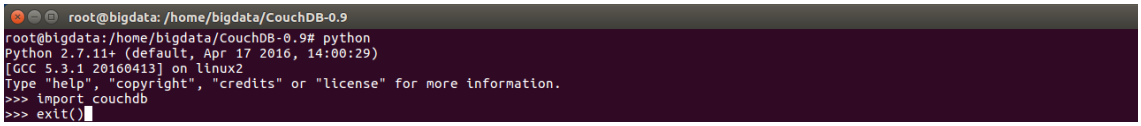


Figura 4.155: Importación de CouchDB en python.

- Finalmente es necesario importar tweepy, de esta manera el sistema operativo cuenta con todos los requisitos para poder recolectar los Tweets. En la siguiente sección se detallarán las claves que genera el API de Twitter para poder acceder como desarrollador.

```
$import tweepy
```

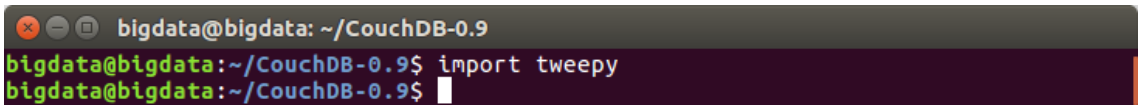


Figura 4.156: Importación tweepy.

Paso 4: Ingreso al API de Twitter.

Para acceder a Twitter como desarrollador es necesario contar con una cuenta de Twitter creada previamente, como ya se había mencionado antes. Una vez que se cuente con este requisito, se podrá acceder a las claves que Twitter genera para proceder al desarrollo del programa en Python.

- Acceder a Twitter Application Management, como se ilustra en la siguiente imagen:

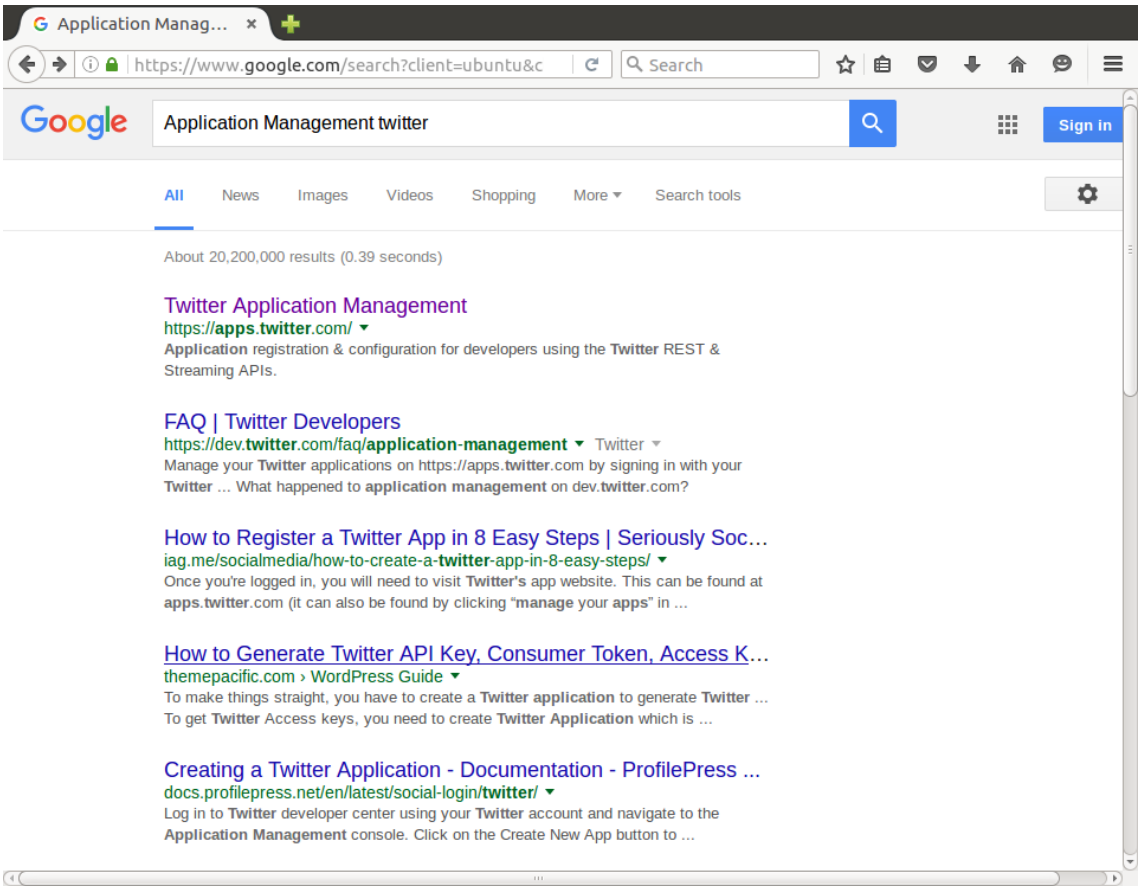


Figura 4.157: Twitter Application Management.

- Ingresar a esta plataforma con la cuenta de Twitter ya creada, al acceder a ella se muestra la siguiente interfaz:

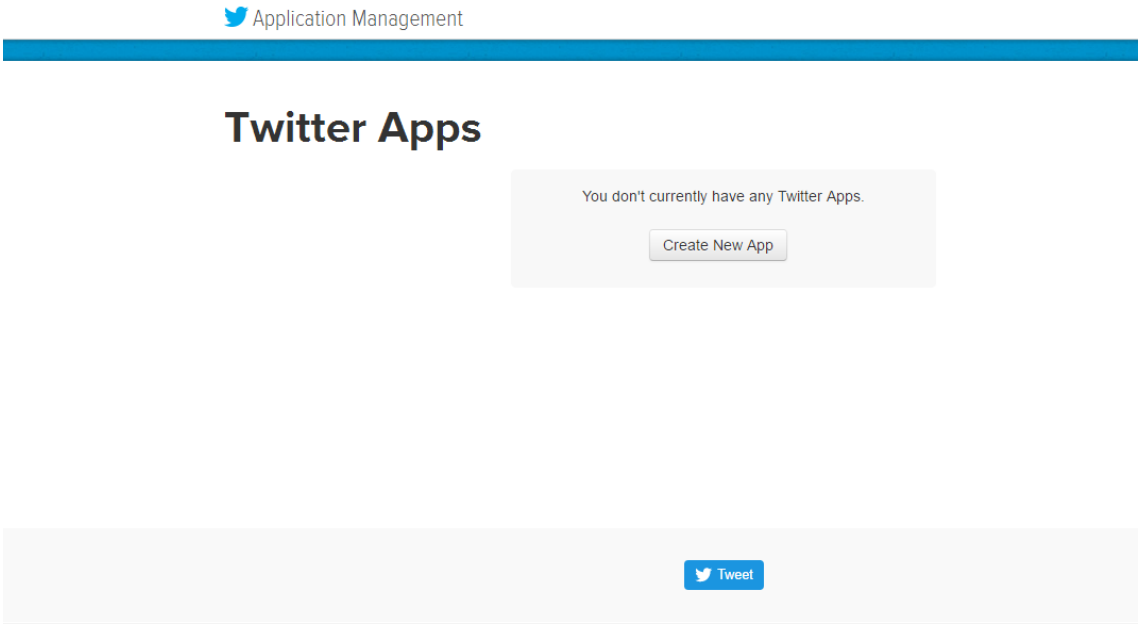


Figura 4.158: Acceso a Application Management.

- Al acceder a esta plataforma se debe dar Click en la opción *Create New App*, lo que significa que se va a crear una nueva aplicación. Desplegándose así lo siguiente:

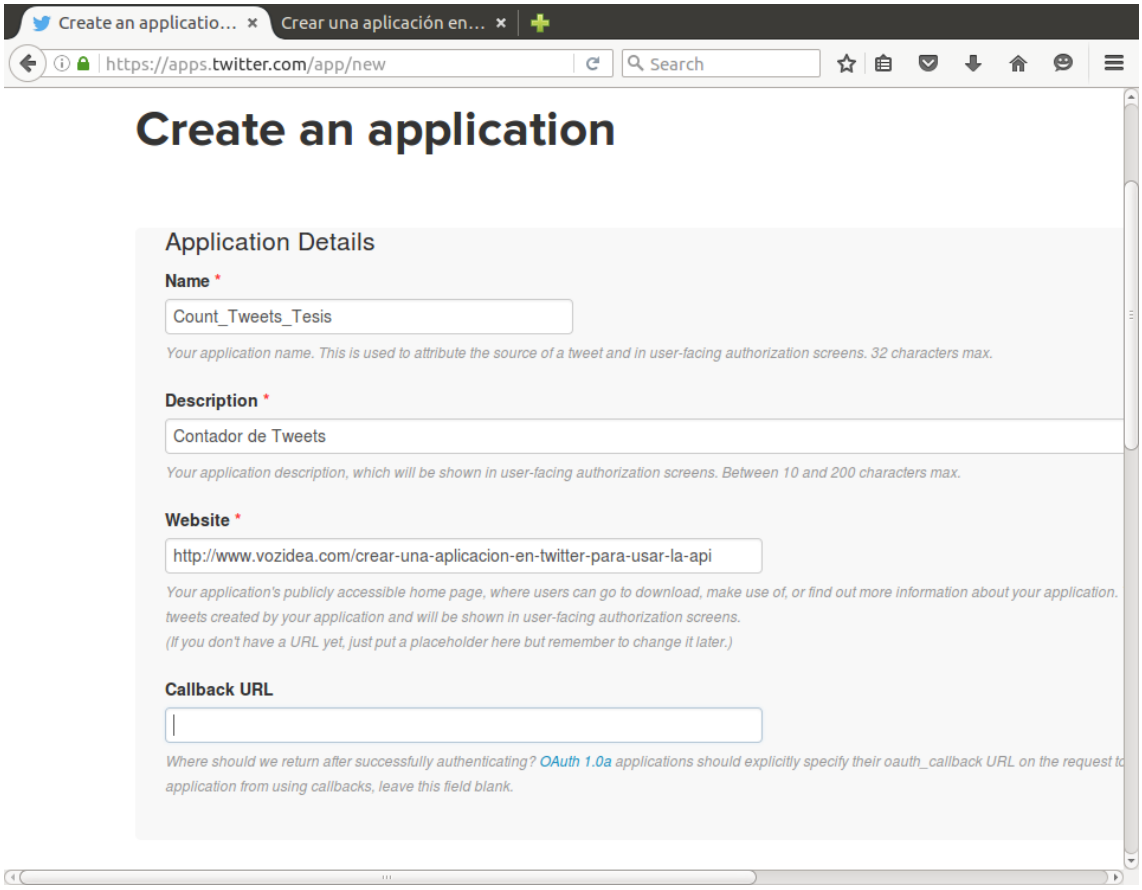


Figura 4.159: Creación de la Aplicación.

Esta imagen muestra la ventana de creación de una nueva aplicación, en la cual se despliegan ciertos campos que son obligatorios llenarlos. Por ejemplo, en el Campo ***Name***, se debe colocar el nombre que llevará la nueva aplicación; en el campo ***Description***, se debe ingresar una rápida descripción acerca de lo que hará la aplicación, en este caso se colocó la descripción de Contador de Tweets, puesto que esta aplicación está orientada a la recolección de los tweets; en el Campo ***Website***, se debe colocar una dirección web cualquiera pero que sea válida, si se contara con una página web propia entonces se colocaría el URL de dicha página, en este caso se colocó la dirección de una página web que estuviera en funcionamiento y el campo ***Callback URL*** no es necesario llenarlo así que se lo puede dejar en blanco. Finalmente, se deben aceptar los términos y condiciones y dar click en la opción ***Create your Twitter Application.***

- Se despliega así la siguiente ventana:

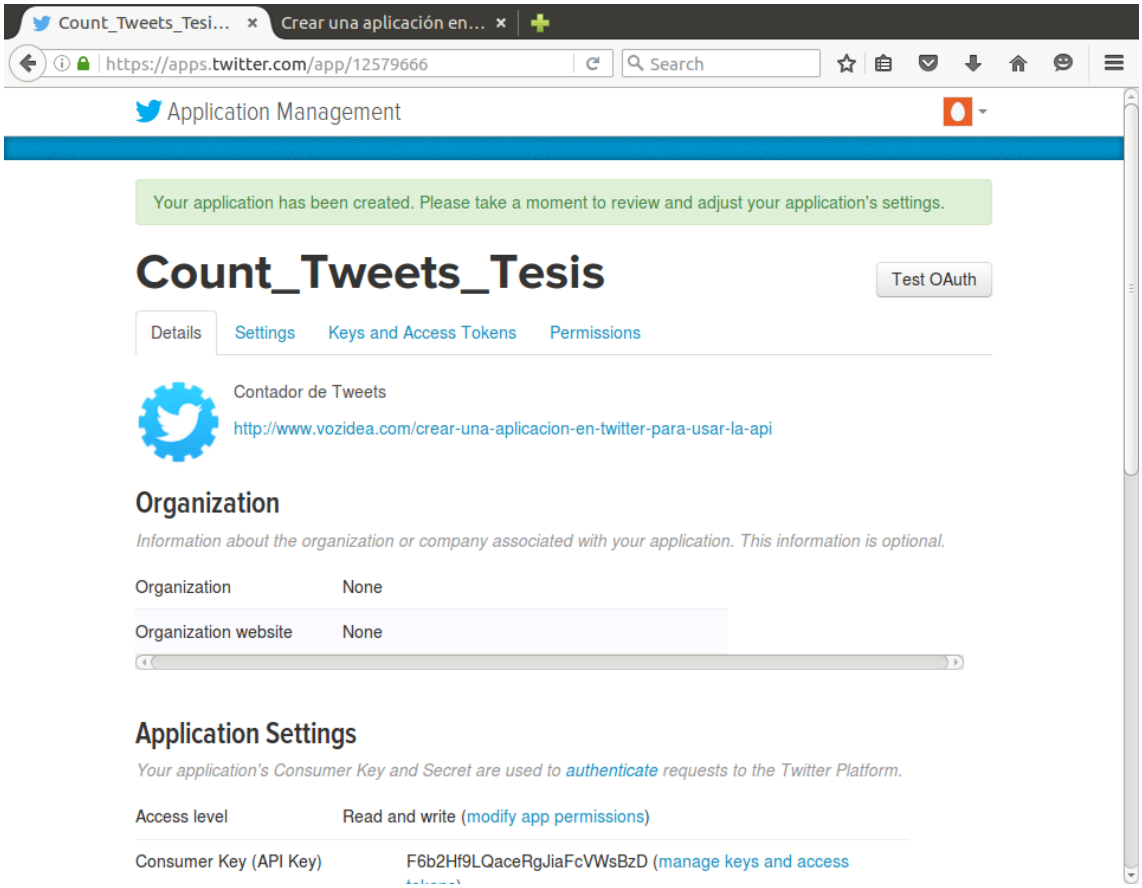


Figura 4.160: Claves generadas por Twitter.

En esta ventana se muestran los datos de creación de la aplicación así como el acceso a las claves que proporciona twitter para la creación de las nuevas aplicaciones, para acceder a ellas es necesario dar click a la opción ***manage keys and access tokens***, tal como se muestra en la siguiente imagen:

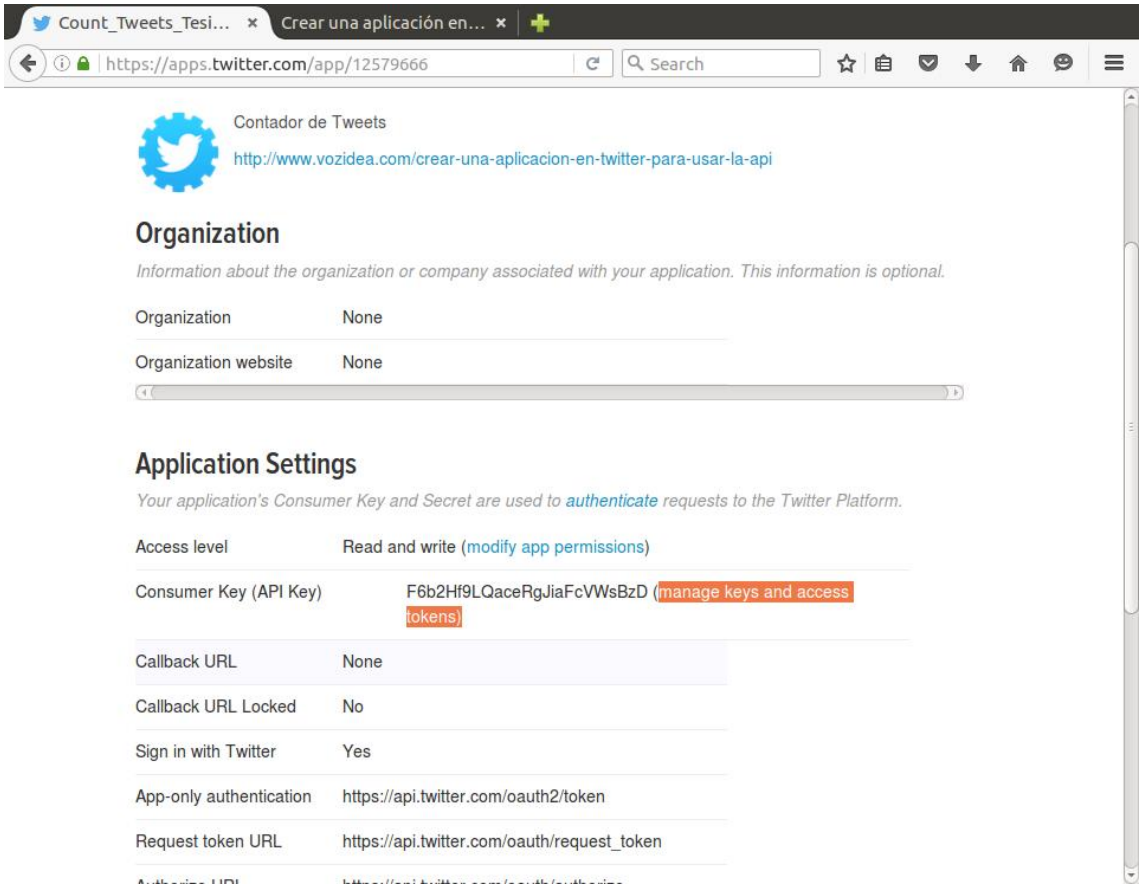


Figura 4.161: Claves generadas por Twitter 2.

- Se despliega la ventana de claves que twitter proporciona:

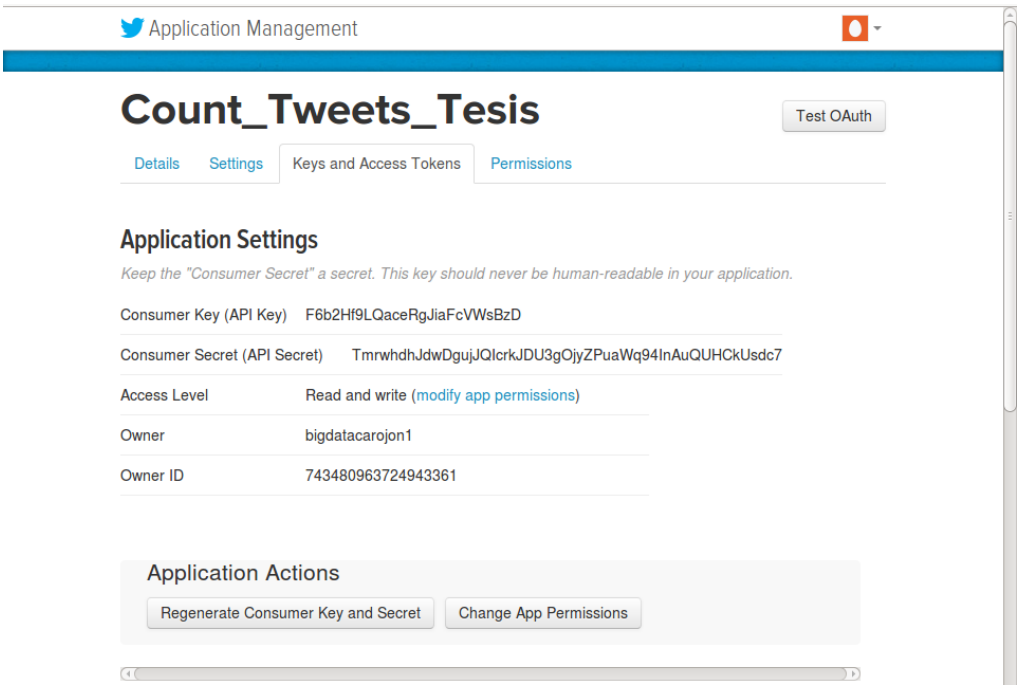


Figura 4.162: Claves como Consumer Key, y Consumer Secret.

Estas claves son necesarias al momento de escribir el código que permitirá la recolección de tweets.

Paso 5: Creación de la Base de datos.

En esta sección se debe crear el nombre de la base de datos con la que se va a trabar, para esto se debe acceder al navegador y digitar la siguiente dirección.

http://127.0.0.1:5984/_utils/

- Se presenta una interfaz como la siguiente, en ella se debe seleccionar la opción *Create Database.*
- Se debe digitar el nombre de la base de datos, en este caso se llamará “quito” en vista de que contendrá los Tweets de Quito. A continuación se muestran las figuras que contienen este proceso.

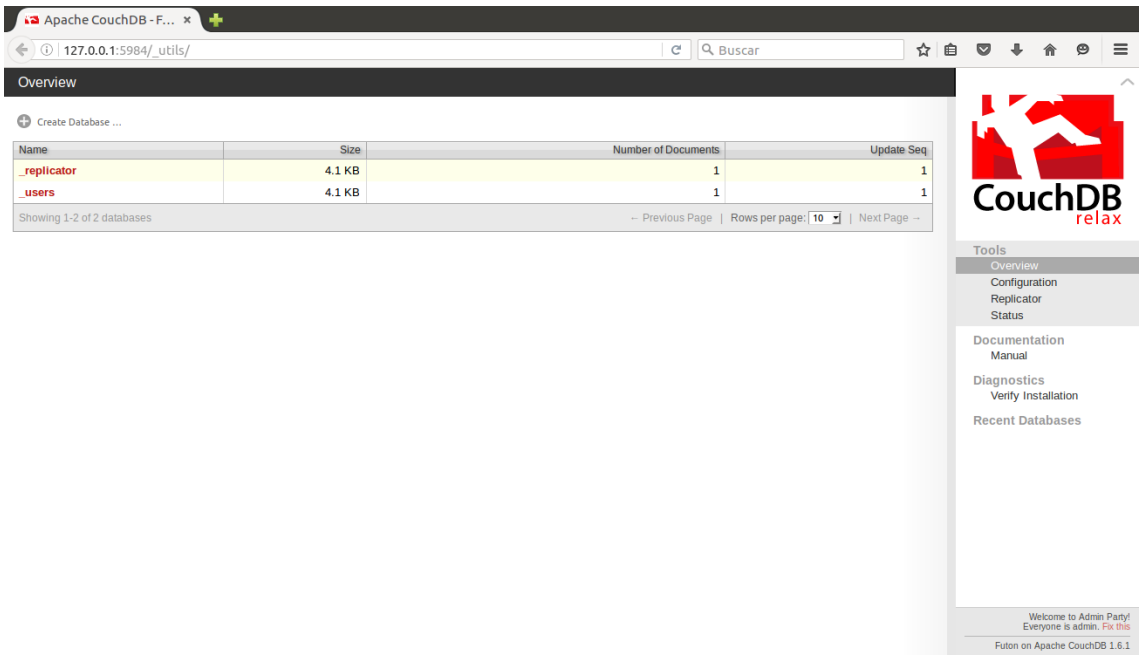


Figura 4.163: Interfaz de CouchDB, creación de nueva base de datos.

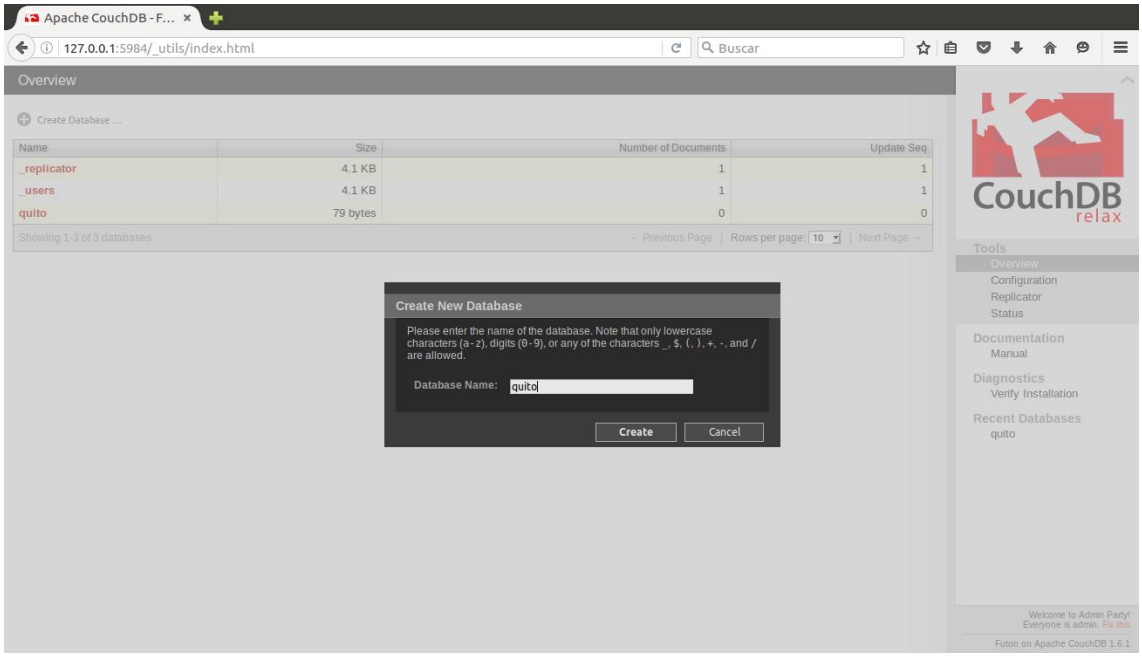


Figura 4.164: Creación de base de datos denominada quito.

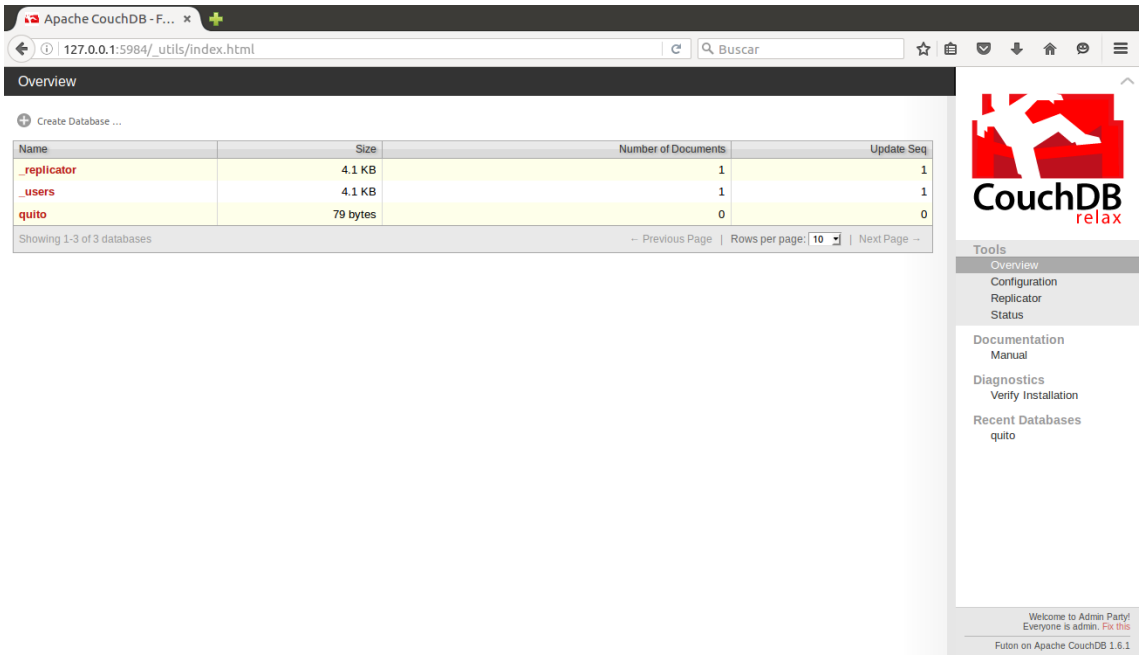


Figura 4.165: Verificación de base de datos.

Paso 6: Creación del Código en Python.

El desarrollo de esta aplicación se la puede realizar en cualquier IDE que soporte Python o se lo puede desarrollar en cualquier gestor de texto. A continuación se presenta el código y el funcionamiento del mismo.

- Las claves que fueron analizadas en pasos anteriores deben ser pegadas en cada una de las secciones, como *Consumer Key API, Consumer Secret API, Access Token, Access Token Secret.*

tweets.py

```
import couchdb
from tweepy import Stream
from tweepy import OAuthHandler
from tweepy.streaming import StreamListener
import json

###API ########################
ckey = "F6b2Hf9LQaceRgJiaFcVWsBzD"   #  Consumer Key API Key
csecret = "TmrwhdhJdwDgujJQIcrkJDU3gOjyZPuaWq94InAuQUHCkUsdc7"  #Consumer Secret API Sec
atoken = "743480963724943361-jZrXFB3wfOmIpjycQZs1IHN4GXqkTkb" #Access Token
asecret = "R2Ea1FU1jYx5OttpUMyWr7fYlBfZMhNCkCFQW5PhjMeNS"  #Access Token Secret
#####################################
#Recolección de los Tweets y almacenamiento
class listener(StreamListener):

    def on_data(self, data):
        dictTweet = json.loads(data)
        try:
            dictTweet["_id"] = str(dictTweet['id'])
            doc = db.save(dictTweet)
            print "SAVED" + str(doc) +"=>" + str(data)
        except:
            print "Already exists"
            pass
        return True

    def on_error(self, status):
        print status

auth = OAuthHandler(ckey, csecret)
auth.set_access_token(atoken, asecret)
twitterStream = Stream(auth, listener())

'''========couchdb'=========='''
server = couchdb.Server('http://localhost:5984/')   #Dirección de acceso a CouchDB
try:
    db = server.create('quito') #Base de datos creada con el nombre, en este caso quito
except:
    db = server['quito']

'''===============LOCATIONS=============='''
#Coordenadas delimitadas.
#La latitud sur, latitud norte, longitud oeste, longitud este
twitterStream.filter(locations=[-78.610783,-0.354993,-78.364105,0.00103])
```

Importación de librerías, utilización de tweepy que fue instalada en pasos anteriores.

Claves generadas por Twitter

Recolección de Tweets.

Conexión a CouchDB y acceso a la base de datos creada previamente.

Zona geográfica para ser analizada.

Figura 4.166: Código fuente desarrollado en Python, llamado tweeps.py.

Se recomienda recolectar las coordenadas con la estructura siguiente, latitud sur, latitud norte, latitud oeste, latitud este. Existen varios páginas web que permiten la identificación de coordenadas, para este caso práctico se utilizó **Boundingbox**. A continuación se muestra el funcionamiento del mismo.

- Delimitar la sección a analizar, de esta manera se genera las latitudes en cada uno de los puntos, finalmente estas latitudes deben ser copiadas en el código.

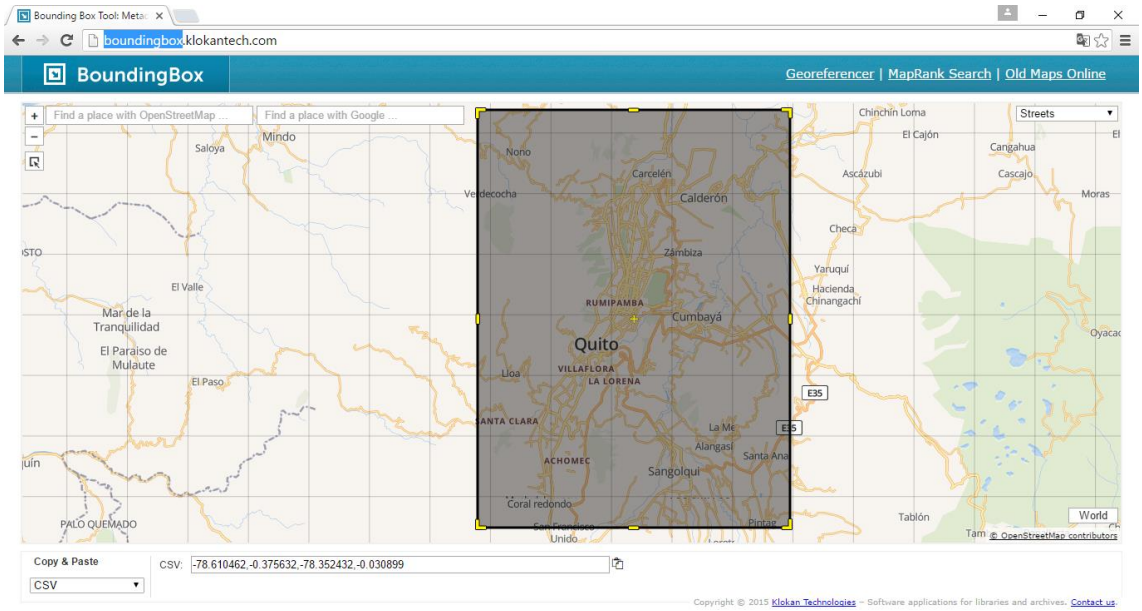


Figura 4.167: Boundingbox, presentación de coordenas.

Este código debe ser guardado en cualquier carpeta del sistema, en este caso se almacenó en Documentos, con el nombre *tweets.py.*

Paso 7: Ejecución del programa.

Finalmente, en esta sección se debe ejecutar el programa desarrollado y verificar como se almacenan los datos. Para esto es necesario acceder a la terminal y continuar con los siguientes pasos:

- Acceder a la carpeta donde está el programa desarrollado.

```
$cd /home/bigdata/Documentos/
```

- Verificar si existe el archivo, para esto digitar el siguiente comando.

```
$ls
```

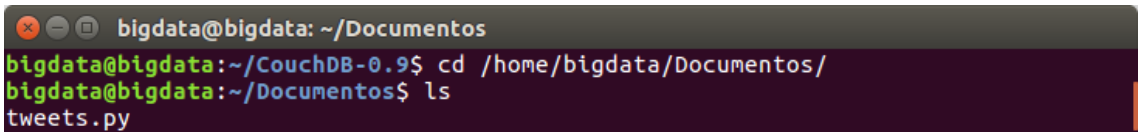


Figura 4.168: Acceso a la carpeta que contiene el código.

- Correr el programa con la siguiente instrucción. Es importante no cerrar la terminal, ya que ésta permite la ejecución del programa, si se llegara a cerrar la terminal la recolección de Tweets terminaría.

```
#python tweets.py
```

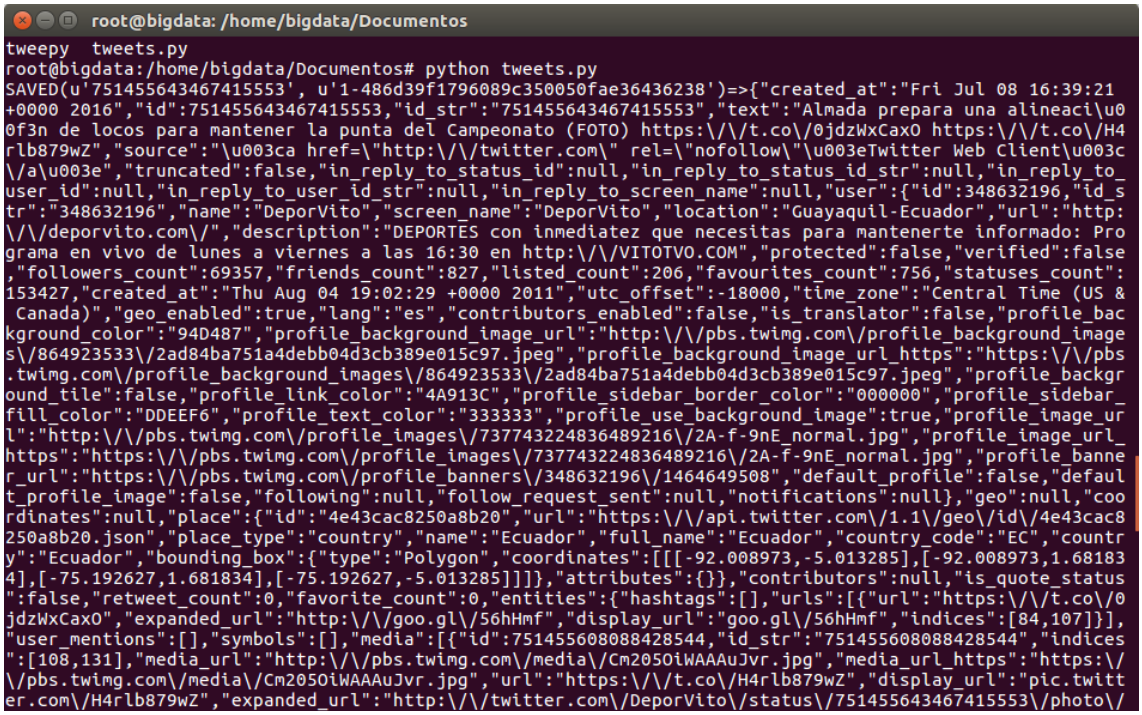


Figura 4.169: Ejecutar tweets.py.

Posteriormente, en la terminal aparecerán varias estructuras que contienen la información de los Tweets. De esta manera se van almacenando los datos en la base de datos NoSQL, en formato JSON.

Para verificar el almacenamiento se debe dirigir al explorador y acceder a la base de datos que fue creada.

Para este caso práctico, la recolección de Tweets fue alrededor de dos días, con esto se puedo obtener alrededor de 112.4 MB de información y 15421 Tweets en el ciudad de Quito. La estructura que maneja CouchDB se puede visualizar en la figura ……. , en ésta aparecen el nombre de usuario, la fecha en la que realizó el Tweet y, además una gran variedad de contenido como el texto, Hashtag, latitud y longitud de donde se realizó esta transacción.

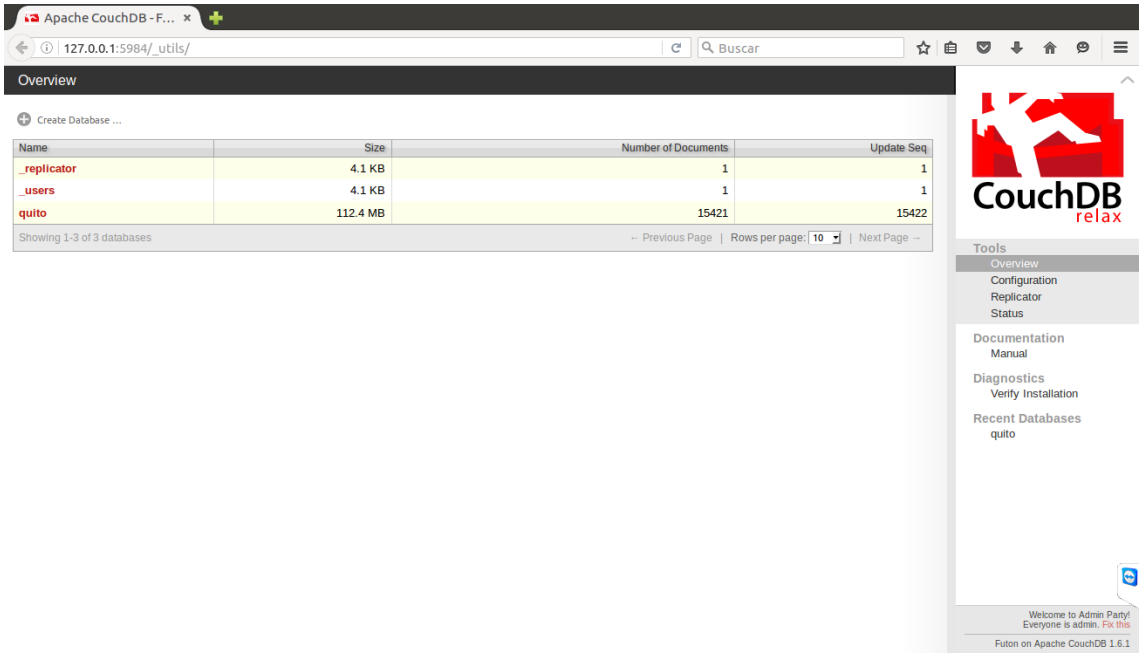


Figura 4.170: Base de datos con alrededor de 15421 Tweets.

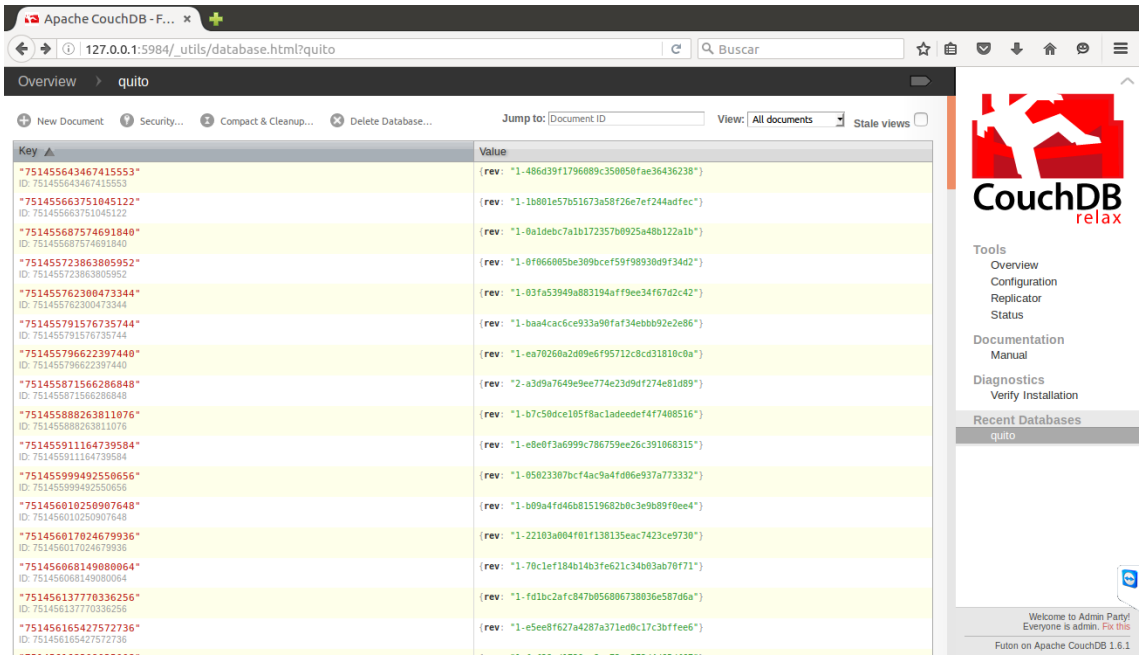


Figura 4.171: Ilustración del almacenamiento.

En este caso muy particular la persona que realizó esta transacción, compartió el siguiente mensaje "*Ya no me acuerdo como es, verte así, muy feliz, una vez..*" y no compartió ningún hashtag, Url, o símbolos en particular.

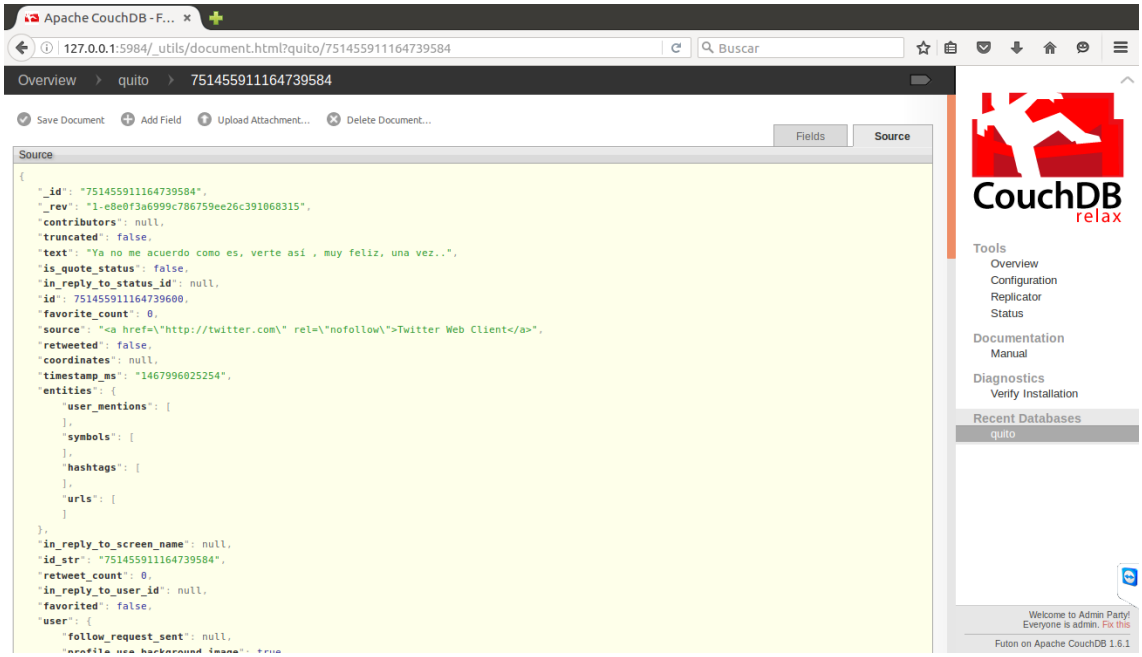


Figura 4.172: Información tweet registrado.

Además se puede visualizar el país e incluso mostrar coordenadas, donde la persona realizó el Tweet.

Figura 4.173: País donde fue emitido el tweet.

CAPITULO V: Metodología de Proceso Enseñanza de Big Data.[13]

El objetivo de este capítulo es elaborar una serie de matrices de planificación a nivel general sobre la cátedra de Big Data, que podrían servir como una guía de apoyo para los profesores que deseen utilizarlas para dictar dicha materia.

Descripción de la Materia:

Los grandes volúmenes de datos, junto al análisis de los mismos son el tronco sobre el cual se desarrollan nuevas estrategias de extracción de valiosa información, así, esta materia pretende explicar y dar a conocer conceptos basados en el procesamiento de grandes cantidades de datos que vienen en distintos formatos y a gran velocidad, haciendo énfasis en las herramientas y tecnologías que facilitan el manejo de datos para el proceso de análisis y extracción de información, lo que es en sí hacer Big Data. De esta manera, se da a conocer a los alumnos las posibles herramientas que pueden ser utilizadas al momento de desarrollar casos prácticos de Big Data, con lo cual se busca que los alumnos sean capaces de usar este conocimiento para el desarrollo de nuevos casos, problemas, investigaciones, etc., que sean útiles en su formación personal y profesional.

Objetivo General de la Materia:

Explicar a los estudiantes los conocimientos necesarios sobre Big Data partiendo de conceptos relevantes, herramientas y tecnologías que expongan su utilidad.

[13] Matrices de planificación didáctica tomadas de: (Alomoto Talavera, 2012)

5.1. Matriz de planificación didáctica acera de lo que es Big Data.

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
ESCUELA DE SISTEMAS
UNIDAD DIDÁCTICA N° 1

1. **DATOS INFORMATIVOS**

Duración: Una semana

2. **OBJETIVO:** Proporcionar a los estudiantes conceptos iniciales acerca de Big Data para que se familiaricen con esta tendencia y sea más fácil el aprendizaje de nuevos conceptos durante el desarrollo y continuación de la Cátedra de Big Data.
3. **TÍTULO DEL CAPÍTULO O UNIDAD:** ¿Qué es Big Data?

CONTENIDO DE APRENDIZAJE	ESTRATÉGIAS METODOLOGICAS	RECURSOS DIDÁCTICOS	ESTÁNDARES DE EVALUACIÓN
Conceptual ✓ Definición ✓ Las 4 V's de Big Data ✓ Tipo de Datos ✓ Historia/ Evolución ✓ Importancia ✓ Predicción de Eventos con Big Data ✓ Conceptos Generales del Capítulo **Procedimental** ✓ Leer Capítulo 1 del presente trabajo de disertación de grado, páginas 7 a la 16 ✓ Ejercitar la lectura en otros libros acerca de Big Data por ejemplo: "Big Data in History" de Patrick Manning. "Big Data Analytics" de Frank Ohlhorst. **libros disponibles en la Biblioteca Virtual de la Puce en ebray.*	**Experiencia Concreta** ✓ Ver videos acerca de la historia de Big Data. **Conceptualización** ✓ Definir en propias palabras lo que significa Big Data. **Aplicación** ✓ Realizar un informe con las ideas principales y los conocimientos adquiridos en el primer capítulo.	✓ Textos de apoyo. ✓ Capítulo 1 del presente trabajo de disertación de grado, páginas 7 a la 16. ✓ Videos Educativos e Informativos. ✓ Exposiciones que cubran el contenido por parte del profesor y alumnos. ✓ Debate sobre posibles eventos que puedan ser pronosticados utilizando Big Data. ✓ Tareas en equipo.	Al finalizar el capítulo o unidad los alumnos podrán: ✓ Definir con sus propias palabras lo que es Big Data. ✓ Tener una visión clara de lo que significa hacer Big Data. ✓ Conocer la importancia de Big Data en estos días. ✓ Entender como inició el Big Data.

5.2. Matriz de planificación didáctica acera de las generalidades de Big Data.

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
ESCUELA DE SISTEMAS
UNIDAD DIDÁCTICA N° 2

1. **DATOS INFORMATIVOS**

Duración: Dos semanas

2. **OBJETIVO:** Proporcionar a los estudiantes conceptos más objetivos y propios de Big Data.
3. **TÍTULO DEL CAPÍTULO O UNIDAD:** Generalidades de Big Data.

CONTENIDO DE APRENDIZAJE	ESTRATÉGIAS METODOLOGICAS	RECURSOS DIDÁCTICOS	ESTÁNDARES DE EVALUACIÓN
Conceptual ✓ Áreas de Big Data. ✓ Paradigmas de Big Data. ✓ Conceptos generales del capítulo. **Procedimental** ✓ Leer Capítulo 2 del presente trabajo de disertación de grado, páginas 18 a la 37. ✓ Ejercitar la lectura en otros libros acerca de Big Data por ejemplo: "Big Data + Analítica Web" de Tomás Baiget, Carlos Tejada Artigas, Natalia Arroyo Vázquez. "Big Data" de Hrushikesha Mohanty, Prachet Bhuyan, Deepak Chenthati. **libros disponibles en la Biblioteca Virtual de la Puce.*	**Experiencia Concreta** ✓ Ver videos acerca casos reales y experiencias con Big Data. **Conceptualización** ✓ Definir en propias palabras las áreas más importantes de Big Data y poder explicar los paradigmas. **Aplicación** ✓ Realizar un informe con las ideas principales y los conocimientos adquiridos en el segundo capítulo, dar ejemplos de Big Data reales.	✓ Textos de apoyo. ✓ Capítulo 2 del presente trabajo de disertación de grado, páginas 18 a la 37. ✓ Videos Informativos. ✓ Aprendizaje autónomo. ✓ Aprendizaje basado en problemas donde los alumnos puedan aplicar y resolver ejercicios de MapReduce. ✓ Método de preguntas acerca del contenido.	Al finalizar el capítulo o unidad los alumnos podrán: ✓ Conocer cuáles son las áreas de mayor importancia de Big Data. ✓ Aprender los paradigmas de Big Data. ✓ Conocer cómo funciona MapReduce.

5.3. Matriz de planificación didáctica acera de la enseñanza de las tecnologías y herramientas de Big Data.

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
ESCUELA DE SISTEMAS
UNIDAD DIDÁCTICA N° 3

1. **DATOS INFORMATIVOS**

Duración: Una semana

2. **OBJETIVO:** Lograr que los estudiantes conozcan cuáles son las principales tecnologías de Big Data y qué plataformas y herramientas permiten trabajar con Big Data.
3. **TÍTULO DEL CAPÍTULO O UNIDAD:** Herramientas y Tecnologías de Big Data.

CONTENIDO DE APRENDIZAJE	ESTRATÉGIAS METODOLOGICAS	RECURSOS DIDÁCTICOS	ESTÁNDARES DE EVALUACIÓN
Conceptual ✓ Plataformas de Big Data. ✓ Apache Hadoop. ✓ Apache Spark. ✓ Oracle Big Data Appliance. ✓ Introducción a las Tecnologías de Big Data. ✓ Tabla comparativa de las Plataformas. **Procedimental** ✓ Leer Capítulo 3 del presente trabajo de disertación de grado, páginas 40 a la 47. ✓ Ejercitar la lectura en otros libros acerca de las herramientas de Big Data por ejemplo: "Hadoop for Dummies" de Dirk deRoos, Paul Zikopoulos, Roman Melnyk, Rafael Coss. "Fast Data Processing with Spark" de Holden Karau. **libros disponibles en la Biblioteca Virtual de la Puce, en ebray.*	**Experiencia Concreta** ✓ Leer libros que contengan información acerca de las Herramientas mencionadas en el capítulo. **Conceptualización** ✓ Identificar las herramientas más óptimas para el desarrollo de Big Data. **Aplicación** ✓ Realizar un informe con las ideas principales y los conocimientos adquiridos en el tercer capítulo y explicar qué herramienta es la más óptima para realizar Big Data según el alumno.	✓ Textos de apoyo. ✓ Capítulo 3 del presente trabajo de disertación de grado, páginas 40 a la 47. ✓ Exposiciones que cubran el contenido por parte del profesor y alumnos. ✓ Debate acerca de las herramientas de Big Data propuestas. ✓ Tareas en equipo. ✓ Método de preguntas acerca del contenido. ✓ Aprendizaje autónomo acerca de las herramientas y tecnologías de Big Data.	Al finalizar el capítulo o unidad los alumnos podrán: ✓ Conocer las plataformas en las que se puede desarrollar Big Data. ✓ Identificar las diferencias entre cada una de esas plataformas. ✓ Aprender las tecnologías que utilizan esas plataformas. ✓ Adquirir un criterio propio acerca de cuál es la mejor plataforma a utilizar para desarrollar Big Data.

5.4. Matriz de planificación didáctica acerca de la enseñanza de las tecnologías y herramientas de Big Data.

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
ESCUELA DE SISTEMAS
UNIDAD DIDÁCTICA N° 4

1. **DATOS INFORMATIVOS**

Duración: Dos semanas

2. **OBJETIVO:** Lograr que los estudiantes puedan instalar un ambiente de Big Data y que consigan ejecutar ciertos casos prácticos para conocer cómo funciona el hacer Big Data.
3. **TÍTULO DEL CAPÍTULO O UNIDAD:** Instalación de un ambiente de Big Data y Casos Prácticos.

CONTENIDO DE APRENDIZAJE	ESTRATÉGIAS METODOLOGICAS	RECURSOS DIDÁCTICOS	ESTÁNDARES DE EVALUACIÓN
Conceptual ✓ Instalación de un ambiente de Big Data. ✓ Casos Prácticos. **Procedimental** ✓ Leer Capítulo 4 del presente trabajo de disertación de grado, páginas 48 a la 126. ✓ Ejercitar la lectura en otros libros por ejemplo: "Big Data Analytics with R and Hadoop" de Vignesh Prajapati. "Hadoop Cluster Deployment" de Danil Zburivsky. "Professional Hadoop Solutions" de Boris Lublinsky, Kevin Smith, Alexey Yakubovich. **libros disponibles en la Biblioteca Virtual de la Puce, en ebray.*	**Experiencia Concreta** ✓ Leer libros que contengan información acerca de Hadoop para entender el proceso de instalación. **Conceptualización** ✓ Encontrar alternativas para crear un ambiente de Big Data. **Aplicación** ✓ Realizar un laboratorio siguiendo la guía con los pasos de instalación de Hadoop y otro laboratorio para la ejecución de los casos prácticos.	✓ Textos de apoyo. ✓ Capítulo 4 del presente trabajo de disertación de grado, páginas 48 a la 126. ✓ Método de casos, proponiendo nuevos problemas para ser resueltos con Big Data. ✓ Método de proyectos, en el cual los alumnos realizarán sus propios casos prácticos de Big Data. ✓ Método de preguntas acerca del contenido. ✓ Aprendizaje basado en problemas donde los alumnos puedan aplicar y resolver tareas propuestas por el profesor.	Al finalizar el capítulo o unidad los alumnos podrán: ✓ Instalar un ambiente de Big Data utilizando la herramienta Hadoop. ✓ Ejecutar casos prácticos para conocer cómo funciona el realizar Big Data. ✓ Crear sus propios casos prácticos y realizar pruebas de Big Data.

CAPITULO VI: Conclusiones y Recomendaciones

En este capítulo se detallarán y se plantearán una serie de conclusiones y recomendaciones que fueron adquiridas durante todo este trabajo de disertación.

6.1. Conclusiones

1. A través de las investigaciones y análisis realizados se puede determinar que el trato y procesamiento que se les da a los datos y por ende a la información hoy en día es muy diferente al manejo que se le daba a los datos e información en el pasado, debido principalmente al apogeo tanto de dispositivos electrónicos como al aparecimiento masivo de redes sociales en los últimos años. Se espera además, que en los próximos años la información siga creciendo y aumentando de la misma manera la complejidad en cuanto a su gestionamiento, es por eso que las herramientas de manejo de grandes volúmenes de datos son primordiales para ofrecer una solución a este problema.
2. Al elaborar este trabajo de disertación se logró entender algo que debe quedar muy claro para cualquier persona, institución, empresa, etc. Lo cual es que, no tiene objeto el poder analizar y almacenar grandes volúmenes de datos en muy pocos segundos si al final no se va a tener la capacidad de extraer información valiosa de esos datos, es decir, si no se va a conseguir transformar los datos en conocimiento, puesto que una de las reglas más significativas de Big Data es procesar millones de datos y que éstos al mismo tiempo generen valor y conocimiento.
3. Las plataformas de Big Data al permitir el manejo de datos estructurados y no estructurados, conceptos que fueron explicados en el Capítulo 1, presentan un gran beneficio para la toma de decisiones gracias a la facilidad de manejar todos esos tipos de datos, lo cual proporciona ventajas tanto para la vida profesional como para los diferentes campos de la ciencia, ya que por ejemplo en la medicina al analizar el historial de pacientes y sus síntomas sobre una determinada enfermedad, es más fácil poder prevenir a otros pacientes que podrían presentar el mismo mal; asimismo, en el campo de los negocios, por ejemplo, al analizar los datos generados en el tiempo en cuanto al porcentaje de ganancias y pérdidas de una empresa, se puede predecir con mayor facilidad lo que ocurrirá en los años posteriores. De esta manera, se puede concluir que mediante Big Data y al analizar correctamente los datos extraídos, se pueden tomar mejores decisiones.

4. Durante el proceso de desarrollo de este trabajo, también se pudo ultimar que el montar una infraestructura para Big Data puede resultar un tanto costoso tanto por las herramientas a utilizar como por los servicios que se pueden brindar. Por lo tanto, si se desea empezar a ofrecer servicios de solución para Big Data, antes que tener la infraestructura necesaria se debe contar con personas capacitadas en cuanto al desarrollo de algoritmos de análisis de datos, porque a fin de cuentas las herramientas siguen simplemente instrucciones, pero son las personas las que hacen funcionar a las herramientas y las que con su ingenio y capacidad pueden ofrecer incluso mejores resultados, por ende, el contar con un personal eficiente y preparado en temas de administración es clave.
5. Por otro lado, si se desea contar con herramientas que ofrezcan soluciones para Big Data, se pudo determinar que Hadoop y Spark se encuentran dentro de las principales plataformas que permiten realizar Big Data, debido principiante a que ambas son herramientas de software libre y que soportan el manejo de datos estructurados y no estructurados, además cuentan con sus propias tecnologías permitiendo guardar estos tipos de datos para poder extraer solo aquellos que generen información de gran valor.
6. Al realizar el Capítulo 4 en el apartado 4.1 correspondiente a la instalación de un ambiente de Big Data, se decidió escoger como plataforma a la herramienta Hadoop en un sistema operativo Centos, ya que el objetivo consistía en montar un clúster con tres nodos para determinar cómo se da el manejo y distribución de datos en este tipo de arquitectura, una vez realizada la práctica se pudo concluir que Hadoop es una herramienta útil cuando se desea armar un clúster en un ambiente totalmente distribuido, además que ésta es una poderosa herramienta gracias a la distribución y al paralelismo de sus ejecuciones. Por otro lado, se pudo determinar que mientras más nodos existan dentro del clúster de Hadoop, más rápida será la distribución de datos y en menos tiempo se obtendrán los resultados.
7. Se puede concluir además, que la utilización del sistema operativo Centos no fue la decisión más acertada debido a su complejidad en cuanto a la compatibilidad de versiones tanto con hadoop como con la instalación de CouchDB, prácticas que se encuentran en el Capítulo 4. Se necesitó de una serie de procesos prueba/error, para lograr los objetivos planteados en cuanto a la realización de éstas prácticas. Pero, al mismo tiempo esta serie de dificultades que se presentaron, constituyen una ventaja para nosotros debido a que logramos familiarizarnos con este sistema operativo y conocer más acerca de su funcionamiento y de los comandos claves y más importantes.

6.2. Recomendaciones

1. La acumulación de datos pasados, es decir, tener un historial de datos sigue siendo de vital importancia al momento de realizar análisis o modelos estadísticos y analíticos, por lo cual si se desea tener información en tiempo real y procesar todos esos miles de datos, la mejor solución es la utilización de plataformas de Big Data que permitan analizarlos, manejarlos y procesarlos para adquirir lo datos que generen mayor valor y por lo tanto, valiosa información.
2. Dado que Big Data puede ser una solución para la predicción y prevención de problemas en los negocios, enfermedades, etc., y debido a que las tecnologías de Big Data y sus fuentes de datos están en frecuente evolución. Las personas, empresas y compañías que deseen implementar Big Data, deben estar alerta y reaccionar rápidamente para aprovechar los beneficios que brindan las nuevas tecnologías.
3. Es importante, estudiar y entender cómo funciona el algoritmo de MapReduce, debido a que muchas herramientas de Big Data como por ejemplo Hadoop tienen su base en este algoritmo, entonces si lo que se desea es utilizar plataformas para resolver problemas de Big Data, es clave que las personas estén capacitadas en cuanto al funcionamiento de este algoritmo y que al mismo tiempo, sean capaces de crear nuevos algoritmos para el análisis de datos, ya que como se ha mencionado anteriormente, las herramientas no funcionan por sí solas, son las personas la que ponen en marcha a cualquier herramienta o programa.
4. Por otro lado, a la hora de querer utilizar herramientas de Big Data, es necesario informarse y leer sobre los beneficios y desventajas que tienen cada una ellas, para poder escoger las que más se adapten a las necesidades o a los problemas que se desean resolver.
5. Además, como se ha planteado anteriormente el crear una infraestructura de Big Data puede resultar algo costoso, por lo cual, es necesario conocer con claridad cuáles son los problemas que se pretenden resolver y las necesidades que se desean satisfacer para de esta manera, analizar y escoger la plataforma de Big Data que sea más eficiente, que mejor se adapte y que no involucre costos excesivos.
6. Al utilizar Hadoop como plataforma para la realización de una de las prácticas en el presente trabajo de disertación, es importante recomendar que se debe tener muy en cuenta con qué tipo de sistema operativo se prevé trabajar, puesto que el

funcionamiento y el proceso de instalación de Hadoop y de sus complementos no es igual en todos los sistemas operativos.

7. Se recomienda además que si se desea utilizar la herramienta Hadoop como plataforma para Big Data, se la instale en el sistema operativo con el cual la persona tenga mayor familiaridad y conocimiento, para evitar que el uso y manejo de Hadoop sea haga más complejo. Uno de los sistemas operativos en los cuales se recomienda trabajar para la instalación de esta plataforma es el sistema operativo Ubuntu debido a que la curva de aprendizaje del mismo es relativamente mejor y su compatibilidad con versiones y complementos es más amplia.
8. Utilizar esta Guía Metodológica por los profesores que deseen impartir este concepto de Big Data a los estudiantes, ya que fue creada para este propósito.
9. Para poder trabajar con los ejemplos prácticos que presenta esta Guía se pueden realizar laboratorios en los cuales los estudiantes puedan desarrollar estos ejemplos y practicar en ellos para que al mismo tiempo, puedan proponer nuevos casos que se podrían desarrollar en las diferentes herramientas planteadas en los casos prácticos que esta guía ha desarrollado.
10. Se recomienda además, que esta Guía sea modificable y adaptable a las nuevas necesidades para la enseñanza de este concepto de Big Data, ya que se podrían aumentar nuevos casos y añadir mejoras, lo cual serviría de gran ayuda a los profesores interesados en enseñar Big Data.

Bibliografía

Alberto Lafuente. (3 de Junio de 2007). *Universidad del País Vasco.* Obtenido de http://www.sc.ehu.es/acwlaroa/SDI/Apuntes/Cap1.pdf

Alomoto Talavera, N. M. (2012). *Repositorio Digital, Pontificia Universidad Católica del Ecuador.* Obtenido de http://repositorio.puce.edu.ec/handle/22000/9559

Angel Rios. (Febrero de 2009). *Oracle Data Integrator.* Recuperado el 19 de Abril de 2016, de http://www.oracle.com/ocom/groups/public/@otn/documents/webcontent/317498_esa.pdf

Bécares, B. (16 de Julio de 2014). *Channelbiz.* Obtenido de http://www.channelbiz.es/2014/07/16/oracle-lanza-big-data-sql-una-herramienta-para-analizar-los-grandes-datos/

Brust, A. (2 de Marzo de 2012). *ZDNet*. Obtenido de Big Data Analytics: http://www.zdnet.com/article/mapreduce-and-mpp-two-sides-of-the-big-data-coin/

Camacho, E. (Julio de 2010). *SG Buzz*. Obtenido de Herramientas y Tecnologías: http://sg.com.mx/revista/42/nosql-la-evolucion-las-bases-datos#.ViGHTJen_IU

Cisco. (s.f.). *Visual Networking Index (VNI)*. Recuperado el 12 de Mayo de 2016, de http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html#~vniforecast

Cloudera. (15 de Octubre de 2015). *Cloudera*. Recuperado el 19 de Abril de 2016, de https://www.cloudera.com/products/apache-hadoop/key-cdh-components.html

Datakind. (2015). *Datakind.org*. Obtenido de http://www.datakind.org/

Debitoor. (7 de Noviembre de 2010). *Debitoor*. Obtenido de Definición de Cloud computing: https://debitoor.es/glosario/definicion-cloud-computing

EcuRed. (26 de Marzo de 2016). *EcuRed*. Obtenido de http://www.ecured.cu/Kernel

FICO. (10 de julio de 2013). *El Big Bang de la Analítica*. Obtenido de http://www.fico.com/landing/infographic/The-Analytics-Big-Bang_es.html

Garcia, R. (29 de Noviembre de 2009). *Apache CouchDB: una base de datos NoSQL (Relax)*. Obtenido de http://www.rgnu.com.ar/tag/couchdb/

Heredero, C. d. (2004). *Informática y Comunicaciones en la Empresa.* Madrid: ESIC Editorial. Obtenido de https://books.google.com.ec/books?id=U0MXWtqjxtsC&pg=PA74&dq=definici%C3%B3n+de+cluster+en+inform%C3%A1tica&hl=es&sa=X&redir_esc=y#v=onepage&q=definici%C3%B3n%20de%20cluster%20en%20inform%C3%A1tica&f=false

IBM. (12 de Frebrero de 2014). *IBM Ecuador*. Obtenido de http://www-01.ibm.com/software/ec/data/infosphere/hadoop/que-es.html

Jean-Pierre Dijcks-Oracle. (3 de Abril de 2014). *ORACLE*. Recuperado el 19 de Abril de 2016, de https://blogs.oracle.com/datawarehousing/entry/updated_price_comparison_for_big

Jiménez, C. M. (Diciembre de 2014). Big data. Un nuevo paradigma. *Anales de Mecánica y Electricidad*, 13. Recuperado el 15 de Febrero de 2016, de http://www.revista-anales.es/web/n_29/pdf/10-16.pdf

Jiménez, C. M. (Diciembre de 2014). Big Data. Un nuevo paradigma. *Anales de Mecánica y Electricidad*, 11. Recuperado el 18 de Febrero de 2016, de http://www.revista-anales.es/web/n_29/pdf/10-16.pdf

Leskovec, J., Rajaraman , A., & Ullman, J. (2014). *Datasets, Mining of Massive.* (A. Ramos Ramón, & J. López Dávila, Trads.) Palo Alto, California, Estados Unidos de América. Recuperado el 12 de Febrero de 2016, de http://infolab.stanford.edu/~ullman/mmds/book.pdf

Leskovec, R. a. (s.f.). *Mining Massive Datasets.* Obtenido de The MapReduce Computational Model (22:04): https://d396qusza40orc.cloudfront.net/mmds/lecture_slides/MapReduce2_TheMapReduceComputationalModel.pdf

Leskovec, Rajaraman, and Ullaman. (s.f.). *Mining Massive Datasets.* Obtenido de https://d396qusza40orc.cloudfront.net/mmds/lecture_slides/MapReduce2_TheMapReduceComputationalModel.pdf

Los Angeles Times. (13 de Abril de 2009). *Iving John Good Statistician helped crack Nazi code*. Obtenido de http://www.latimes.com/local/obituaries/la-me-passings13-2009apr13-story.html

Margaret Rouse. (22 de Enero de 2015). *Techtarget*. Obtenido de http://searchdatacenter.techtarget.com/es/definicion/Base-de-datos-relacional

Marr, B. (2015). Big Data Case Study Collection. *Wiley*, 2-29.

Matei Zaharia, V. S. (11 de Febrero de 2014). *Databricks*. (C. Ramos , & J. López, Productores) Recuperado el 31 de Marzo de 2016, de https://databricks.com/spark/about

Mipagerank. (1 de Marzo de 2003). *¿Qué es el PageRank?* Obtenido de http://www.mipagerank.com/?goto=que-es-el-pagerank

Morros, R. S. (Noviembre de 2013). *Universidad Politécnica de Catalunia*. Obtenido de Big Data, Análisis de herramientas y soluciones: http://upcommons.upc.edu/bitstream/handle/2099.1/19855/90807.pdf?sequence=1

Nik Rouda, Senior Analyst and Adam DeMattia, Research Analyst. (Diciembre de 2015). *Oracle.* Recuperado el 19 de Abril de 2016, de http://www.oracle.com/us/technologies/big-data/eng-systems-for-big-data-esg-wp-2852701.pdf

Ohlhorst, F. J. (2012). *Big Data Analytics: Turning Big Data into Big Money.* New Jersey: John Wiley & Sons.

Ohlhorst, F. J. (2012). *Big Data Analytics: Turning Big Data into Big Money.* North Carolina: John Wiley & Sons. Obtenido de https://books.google.com.ec/books?id=09JagAXxSYgC&printsec=frontcover&dq=big+data+analytics&hl=es&sa=X&sqi=2&redir_esc=y#v=onepage&q=big%20data%20analytics&f=false

Oracle. (Diciembre de 2015). *Oracle.* Recuperado el 4 de Abril de 2016, de http://www.oracle.com/us/products/database/nosql/overview/index.html

Oracle. (2016). *Oracle.* Recuperado el 19 de Abril de 2016, de http://www.oracle.com/lad/products/database/big-data-appliance/overview/index.html

Oracle. (s.f.). *Oracle.* Obtenido de http://www.oracle.com/lad/products/database/big-data-appliance/overview/index.html

O'Ryan, R. E. (3 de Diciembre de 2014). *Dice*. (C. Ramos, & J. López, Productores) Recuperado el 31 de Marzo de 2016, de http://insights.dice.com/2014/03/12/apache-spark-next-big-thing-big-data/

Ricardo Barranco Fragoso, IT Specialist for information Management, IBM Software Group . (18 de Junio de 2012). *IBM developerWorks*. Obtenido de https://www.ibm.com/developerworks/ssa/local/im/que-es-big-data/

Ricardo Barranco Fragoso, IT Specialist for Information Management, IBM Software Group México. (18 de 06 de 2012). */www.ibm.com*. Obtenido de https://www.ibm.com/developerworks/ssa/local/im/que-es-big-data/

SAS. (5 de Enero de 2016). *Statistical Analysis System*. Obtenido de “Sistema de Análisis Estadístico”: http://www.sas.com/es_mx/company-information.html

Schmarzo, B. (2013). Apache Hadoop. En B. Schmarzo, *Big Data: Undersatanding How Data Powers Big Business. (Big Data, El poder de los datos)* (V. Gonzáles León, Trad., pág. 212). Madrid, España: ANAYA. Recuperado el 30 de Marzo de 2016

Schmarzo, B. (2013). *Big Data: Understanding How Data Powers Big Business (Big Data, El poder de los datos).* (L. Vicente González, Trad.) Madrid, España: ANAYA.

Shakuntala Gupta, E., & Sabharwal, N. (2015). Pactical MongoDB: Architecting, Developing, and Administering MongoDB. Nueva York: Apress.

Sullivan, D. (8 de Abril de 2014). *Tom's IT PRO*. Obtenido de Getting Started with Hadoop 2.0.

Tablet Army. (2012). Manual Básico. Qué es el Big Data. En *Manual Básico. Qué es el Big Data* (pág. 6). Madrid: Prodigioso Volcán.

The Apache Software Foundation. (26 de Enero de 2016). *Hadoop*. Obtenido de https://hadoop.apache.org/docs/r2.7.2/hadoop-project-dist/hadoop-common/SingleCluster.html#Standalone_Operation

Trujillo, J. C. (2013). *Diseño y explotación de almacenes de datos: conceptos básicos de modelado multidimensional.* Alicante, España: ECU. Obtenido de http://puceftp.puce.edu.ec:2057/lib/pucesp/reader.action?docID=10751536

Trujillo, J. C. (2013). *Diseño y explotación de almacenes de datos: conceptos básicos de modelado multidimensional.* Alicante, España: ECU. Obtenido de http://puceftp.puce.edu.ec:2057/lib/pucesp/reader.action?docID=10751536&ppg=2

Trujillo, J. C. (2013). *Diseño y explotación de almacenes de datos: conceptos básicos de modelado multidimensional.* Alicante: ECU. Obtenido de http://puceftp.puce.edu.ec:2057/lib/pucesp/reader.action?docID=10751536

Twitter. (2016). *Centro de Ayuda Twitter*. Recuperado el 06 de 07 de 2016, de https://support.twitter.com/articles/332061

ANEXOS

ANEXO 1: Manual de Usuario Hadoop.

El clúster de Hadoop consta de cuatro máquinas las cuales están etiquetadas o nombradas de la siguiente manera:

Máquina 1:

Nombre: principal

Contraseña: carojona

IP: 192.168.1.100

Hostname y Dominio: master-node.centos

Máquina 2:

Nombre: secundario1

Contraseña: carojona

IP: 192.168.1.101

Hostname y Dominio: slave1-node.centos

Máquina 3:

Nombre: secundario2

Contraseña: carojona

IP: 192.168.1.102

Hostname y Dominio: slave2-node.centos

Máquina 4:

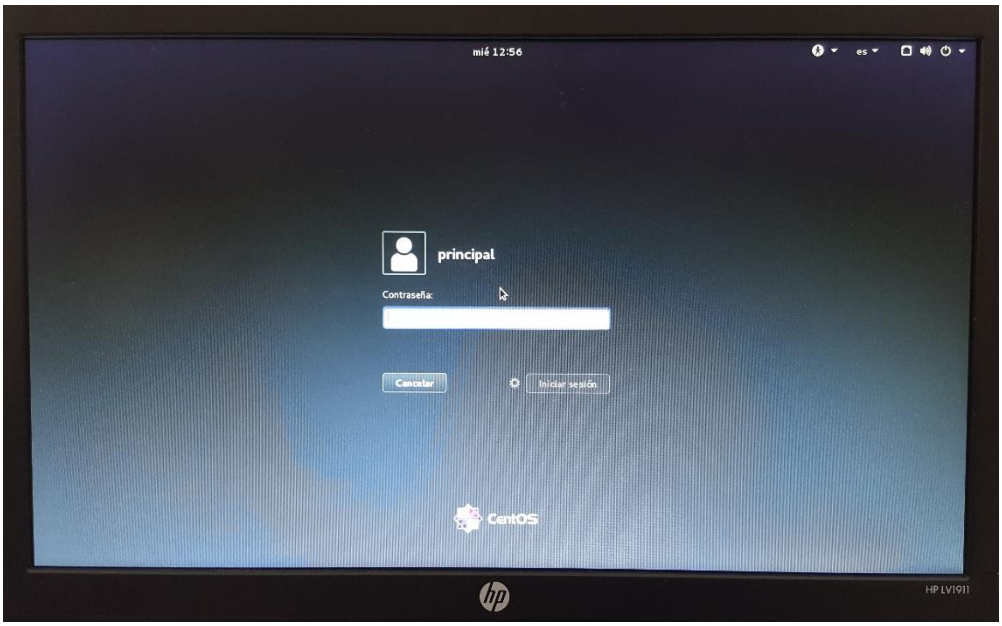
Nombre: secundario3

Contraseña: carojona

IP: 192.168.1.103

Hostname y Dominio: slave3-node.centos

Todas las máquinas constan de la misma contraseña, para el ingreso al sistema.



Una vez que se haya ingresado al sistema, dirigirse a la terminal e ingresar con privilegios de *root*, en vista que la instalación y configuración fue realizada en este usuario.

Una vez ingresado como root, se procede a encender los servicios de Hadoop de la siguiente manera:

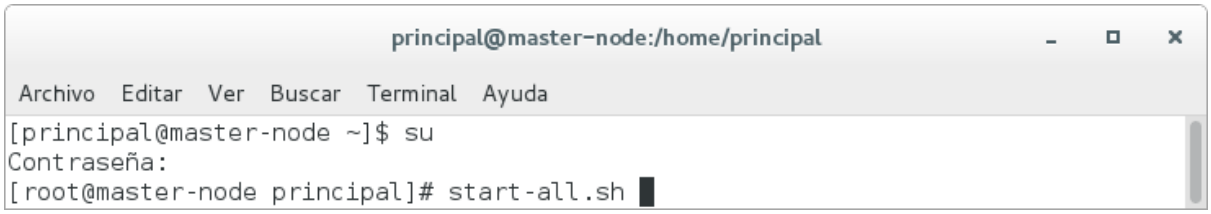
- Para ingresar con privilegios de root digitar el siguiente comando:

```
$su
```

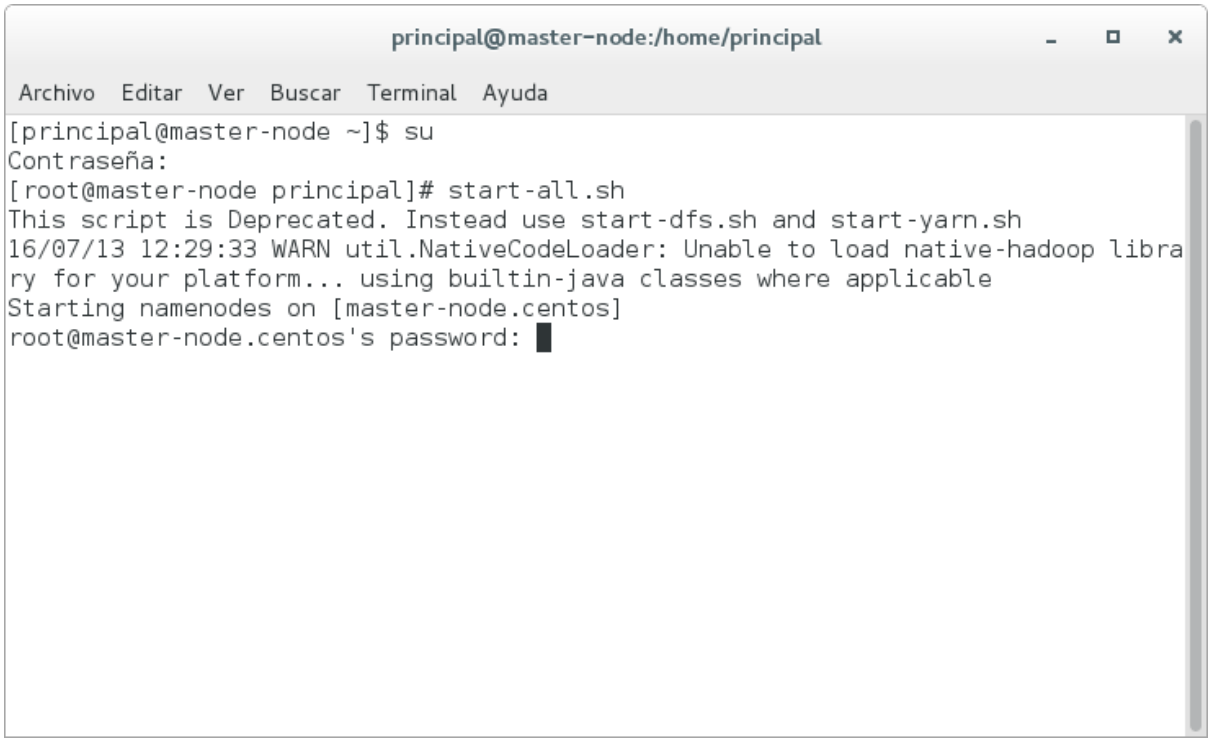
Digitar la contraseña: carojona

- Para encender los servicios de Hadoop, es decir, los demonios del Nodo Master y Nodos Esclavos desde la máquina correspondiente al Nodo Master (principal), digitar el siguiente comando:

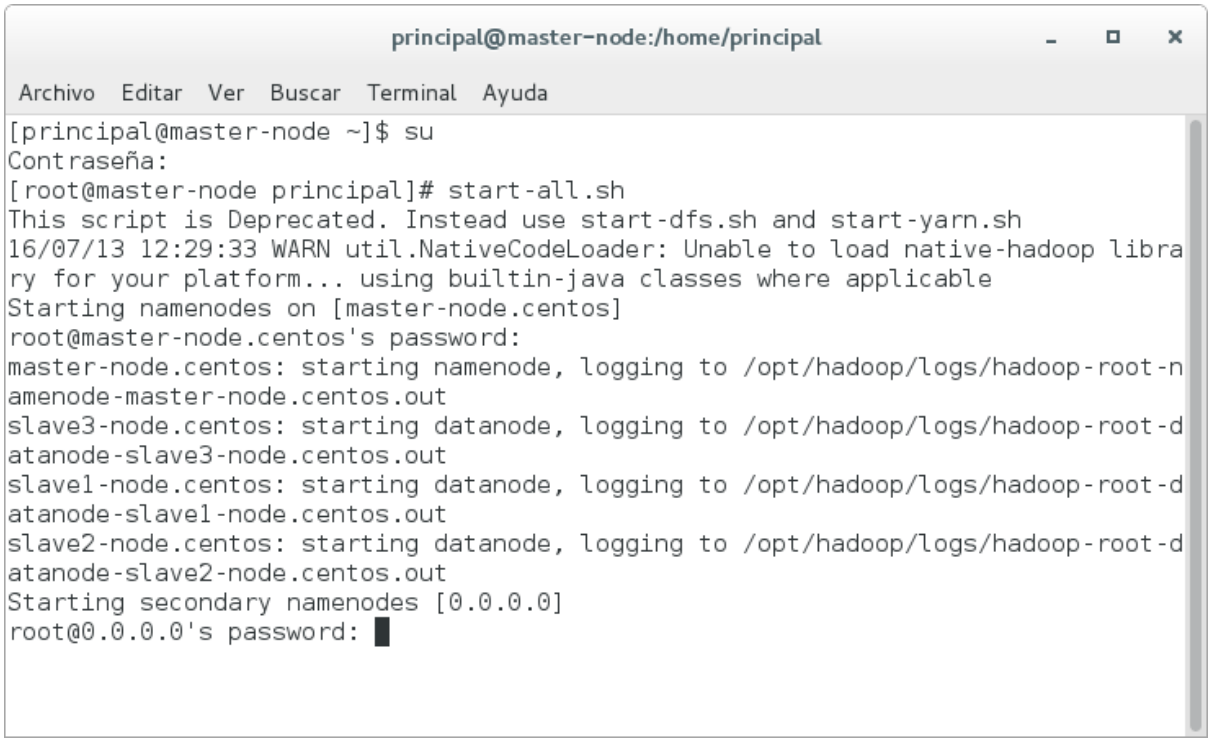
```
#start-all.sh
```



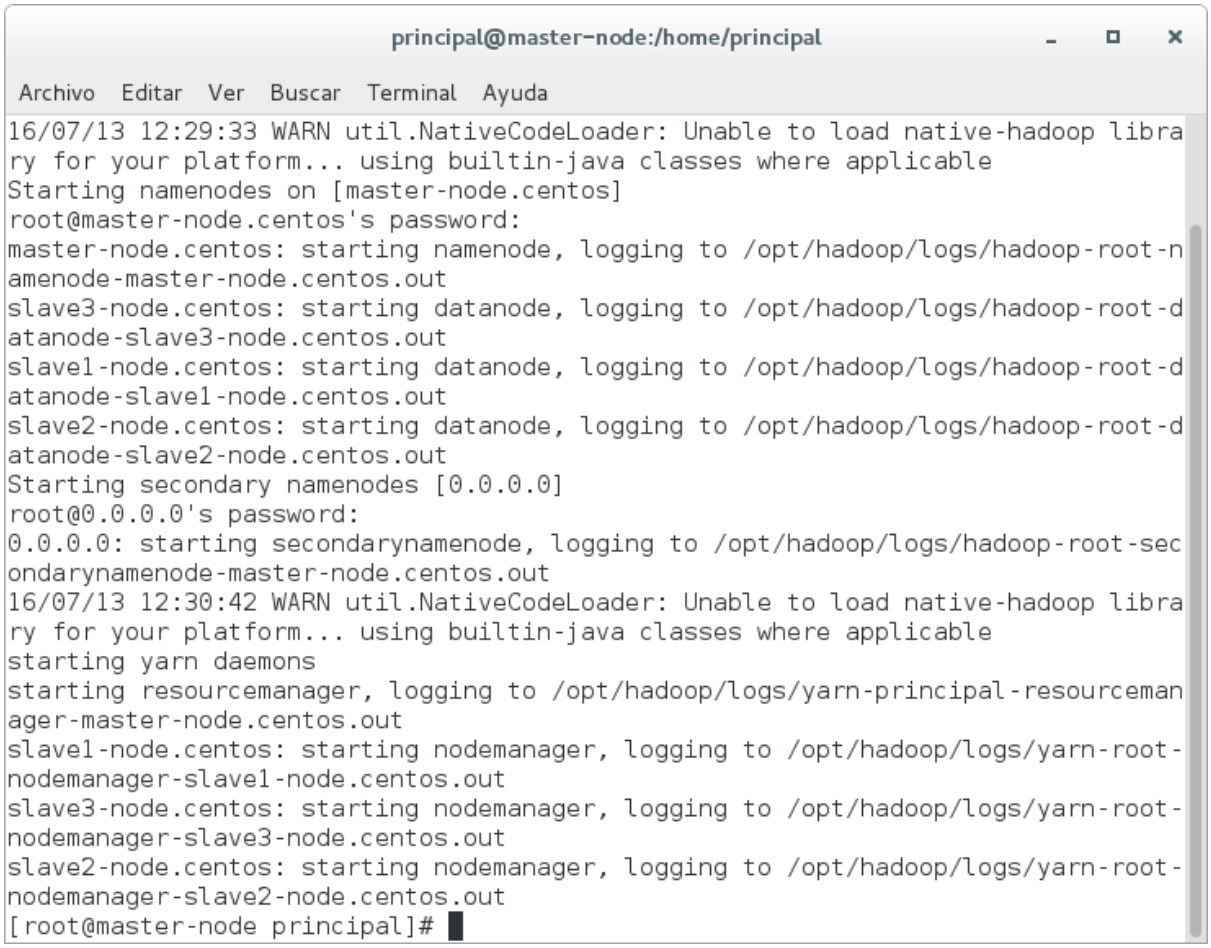
Al encender los servicios de Hadoop, se visualizará lo siguiente:



Como se muestra en la imagen superior, es necesario digitar la contraseña de la máquina correspondiente al host ***master-node.centos,*** la cual es: carojona. Se pide ingresar una contraseña puesto que se está accediendo a la máquina nombrada. Una vez realizado esto se encenderán los servicios como se puede visualizar en la siguiente imagen:



Así se puede observar como los servicios *datanode* se encendieron en cada uno de los Nodos Esclavo. De igual manera, en la imagen superior se muestra que es necesario ingresar una contraseña, ésta es la misma que en el paso anterior, es decir: carojona. Una vez ingresada la contraseña, se encienden los demás servicios como el *nodemanager* y *resourcemanager* tal como se evidencia en la siguiente imagen:



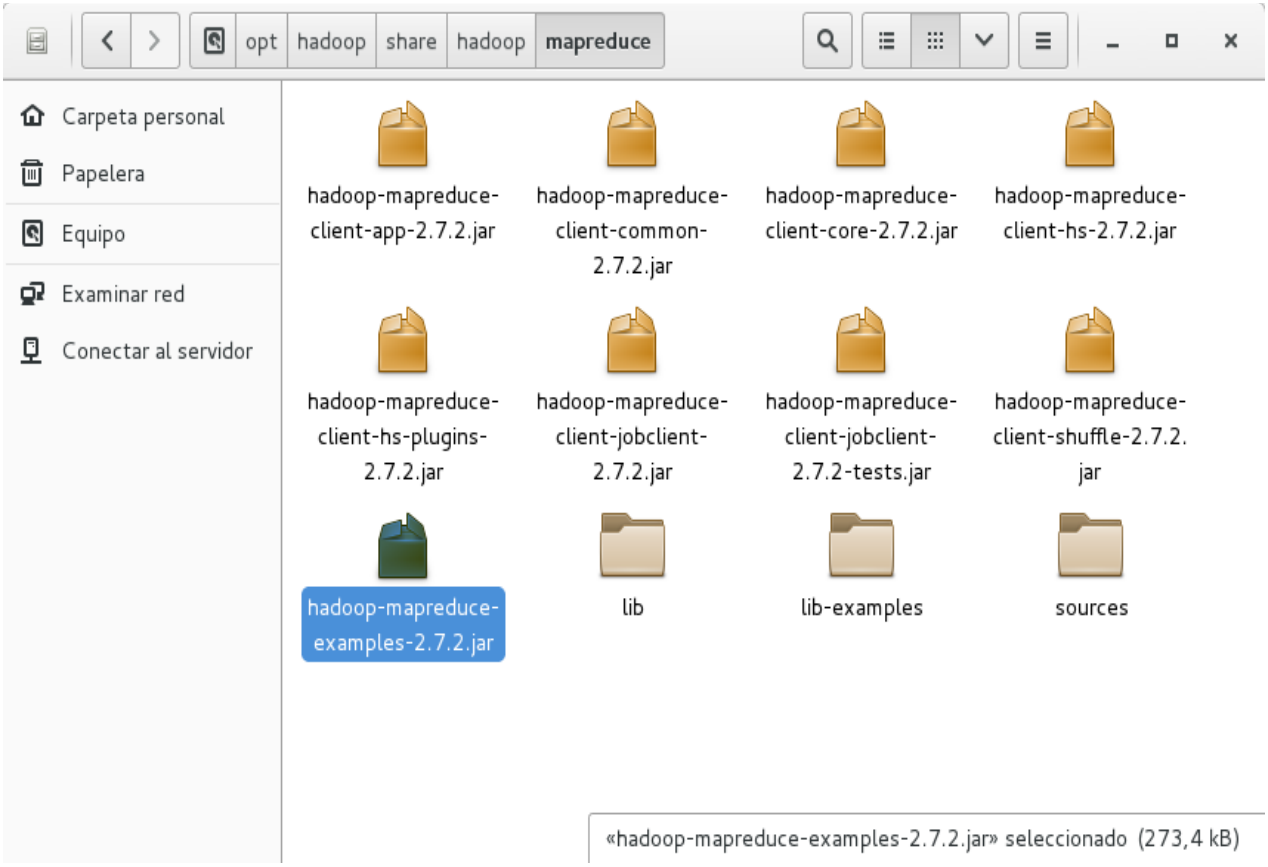
Realizados los pasos anteriores, quedarían inicializados todos los servicios de Hadoop en el Nodo Master y Nodos Esclavos.

De igual manera, si se desea apagar los servicios de Hadoop desde la máquina del Nodo Master, se puede digitar el siguiente comando:

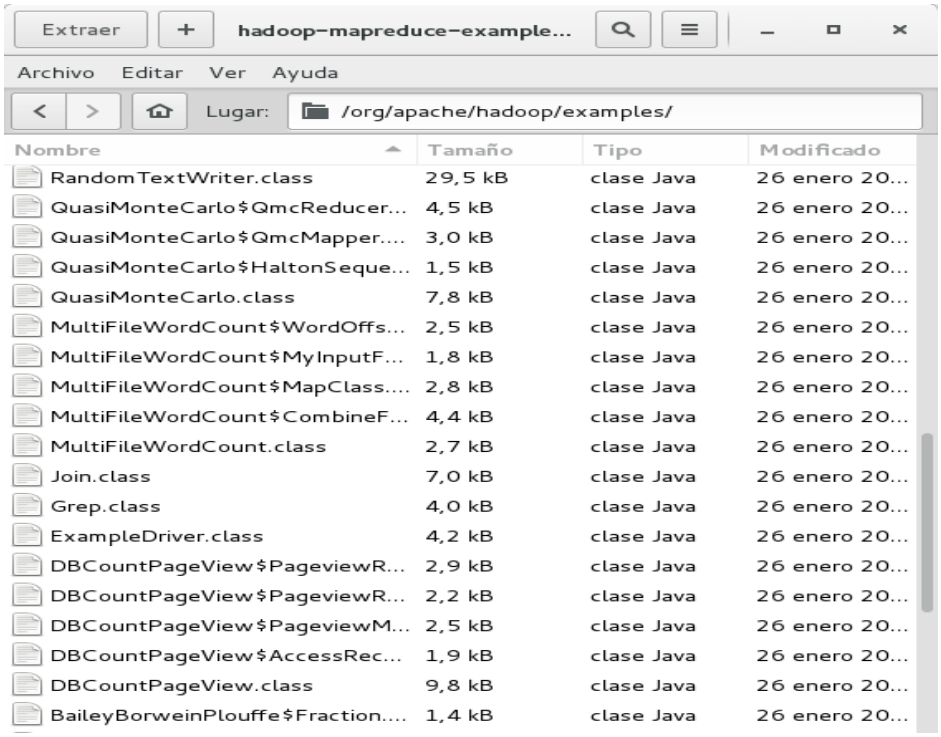
```
#stop-all.sh
```

Hadoop cuenta con algunos ejemplos, tal como se explicó en el Capítulo IV en la Instalación y Configuración de Hadoop. Los ejemplos con los que cuenta esta herramienta se pueden

verificar en la siguiente dirección: *opt/hadoop/share/hadoop/mapreduce*, tal como se muestra a continuación:



El archivo hadoop-mapreduce-examples-2.7.2.jar, cuenta con algunos ejemplos que pueden ser probados como se explicó en el Capítulo IV en la Instalación y Configuración de Hadoop. En la siguiente imagen se pueden observar algunos de los ejemplos con los que cuenta esta herramienta:



Nombre	Tamaño	Tipo	Modificado
RandomTextWriter.class	29,5 kB	clase Java	26 enero 20...
QuasiMonteCarlo$QmcReducer...	4,5 kB	clase Java	26 enero 20...
QuasiMonteCarlo$QmcMapper....	3,0 kB	clase Java	26 enero 20...
QuasiMonteCarlo$HaltonSeque...	1,5 kB	clase Java	26 enero 20...
QuasiMonteCarlo.class	7,8 kB	clase Java	26 enero 20...
MultiFileWordCount$WordOffs...	2,5 kB	clase Java	26 enero 20...
MultiFileWordCount$MyInputF...	1,8 kB	clase Java	26 enero 20...
MultiFileWordCount$MapClass....	2,8 kB	clase Java	26 enero 20...
MultiFileWordCount$CombineF...	4,4 kB	clase Java	26 enero 20...
MultiFileWordCount.class	2,7 kB	clase Java	26 enero 20...
Join.class	7,0 kB	clase Java	26 enero 20...
Grep.class	4,0 kB	clase Java	26 enero 20...
ExampleDriver.class	4,2 kB	clase Java	26 enero 20...
DBCountPageView$PageviewR...	2,9 kB	clase Java	26 enero 20...
DBCountPageView$PageviewR...	2,2 kB	clase Java	26 enero 20...
DBCountPageView$PageviewM...	2,5 kB	clase Java	26 enero 20...
DBCountPageView$AccessRec...	1,9 kB	clase Java	26 enero 20...
DBCountPageView.class	9,8 kB	clase Java	26 enero 20...
BaileyBorweinPlouffe$Fraction....	1,4 kB	clase Java	26 enero 20...

ANEXO 2: Manual de Usuario de Recolección de Tweets (CouchDB).

Para la recolección de tweets, se utilizó una máquina con sistema operativo Ubuntu 16.04, en la cual se encuentran configuradas las librerías que utiliza Python para la recolección de tweets.

Máquina:

Nombre: bigdata

Contraseña: ltic2016

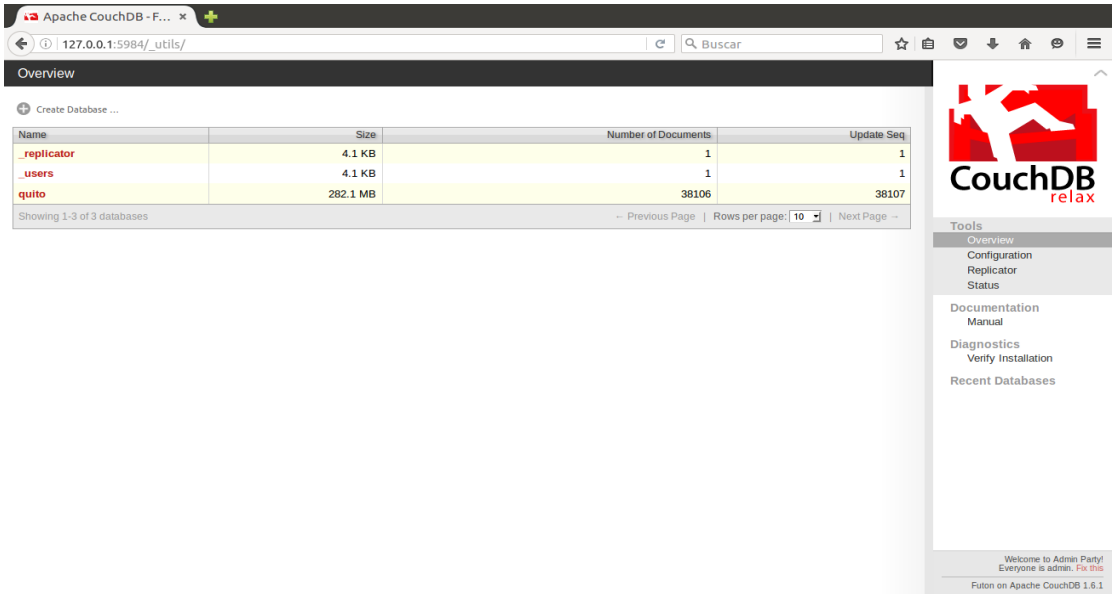
Una vez dentro del sistema, acceder a la terminal y en este caso no es necesario ingresar con privilegios de *root*. En seguida, digitar dentro de la terminal la dirección que contiene el programa desarrollado en lenguaje Python, mismo que servirá para la recolección de los tweets: `cd /home/bigdata/Documentos`/, si se desea comprobar que el archivo se encuentra en esta dirección se puede digitar el comando `ls`.

Ya ubicados en la dirección anterior, se procede a ejecutar el programa digitando lo siguiente: `python tweets.py`. Pasos que se pueden observar en la siguiente imagen:

```
bigdata@bigdata: ~/Documentos
bigdata@bigdata:~$ cd /home/bigdata/Documentos/
bigdata@bigdata:~/Documentos$ ls
tweets.py
bigdata@bigdata:~/Documentos$ python tweets.py
SAVED(u'753273018172989441', u'1-a58be9d36fd7160c2472d5765a41b2e9')=>{"created_a
t":"Wed Jul 13 17:00:57 +0000 2016","id":753273018172989441,"id_str":"7532730181
72989441","text":"\u00bfSabes por qu\u00e9 est\u00e1 china de risa? Porque REGRE
SA #CallaLoca!!! S\u00e1bado 13 de Agosto! Entradas en @teleticketperu https:\/\
/t.co\/lxsI1j7yIl","source":"\u003ca href=\"http:\/\/twitter.com\" rel=\"nofollo
w\"\u003eTwitter Web Client\u003c\/a\u003e","truncated":false,"in_reply_to_statu
s_id":null,"in_reply_to_status_id_str":null,"in_reply_to_user_id":null,"in_reply
_to_user_id_str":null,"in_reply_to_screen_name":null,"user":{"id":222003536,"id_
str":"222003536","name":"Dr. Angulo","screen_name":"Doctor_Angulo","location":"L
ima, Peru","url":"http:\/\/www.tomasangulo.pe","description":"Terapeuta de Parej
a, Sex\u00f3logo y Escritor. reserva de consultas:: 593-6504 RPM #942068818 \/ D
e Lunes a Viernes de 9 a 2 y de 4 a 8 pm.","protected":false,"verified":false,"f
ollowers_count":39642,"friends_count":305,"listed_count":62,"favourites_count":1
3169,"statuses_count":8173,"created_at":"Thu Dec 02 05:37:45 +0000 2010","utc_of
fset":-18000,"time_zone":"Lima","geo_enabled":true,"lang":"es","contributors_ena
bled":false,"is_translator":false,"profile_background_color":"C0DEED","profile_b
ackground_image_url":"http:\/\/abs.twimg.com\/images\/themes\/theme1\/bg.png","p
rofile_background_image_url_https":"https:\/\/abs.twimg.com\/images\/themes\/the
me1\/bg.png","profile_background_tile":false,"profile_link_color":"0084B4","prof
ile_sidebar_border_color":"C0DEED","profile_sidebar_fill_color":"DDEEF6","profil
```

Cuando ya se haya ejecutado el programa es importante no cerrar la terminal, puesto que mientras ésta esté abierta se continuará con la recolección de tweets y si la terminal es cerrada la recolección de los mismos finalizará.

Finalmente, es necesario abrir el explorador para poder acceder a la interfaz de CouchDB, para ello se debe digitar en la barra de direcciones del explorador lo siguiente: http://127.0.0.1:5984/_utils/



Como se muestra en la figura anterior, los tweets han sido guardados en la base de datos ‘quito’, para poder acceder a ellos basta con dar un click en el nombre de la base.